

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-024079

(43)Date of publication of application : 25.01.2002

(51)Int.Cl.

G06F 12/00

(21)Application number : 2001-129926

(71)Applicant : MICROSOFT CORP

(22)Date of filing : 26.04.2001

(72)Inventor : BURD GARY S

COOPER KENNETH B

GUTHRIE SCOTT D

EBBO DAVID S

ANDERS MARK T

PETERS TED A

MILLET STEPHEN J

(30)Priority

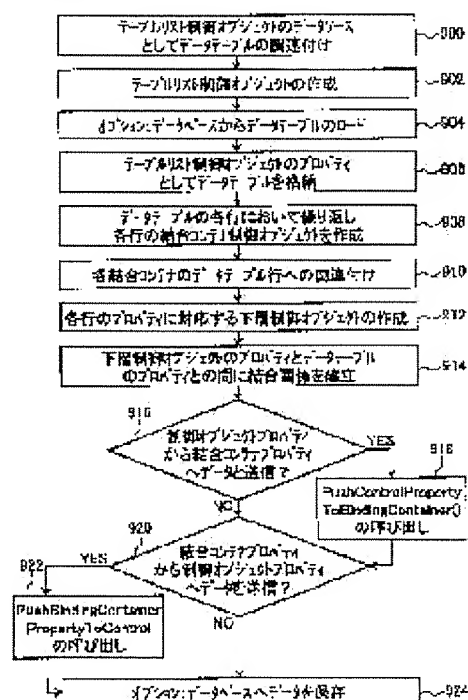
Priority number : 2000 573656 Priority date : 18.05.2000 Priority country : US

(54) DATA JOINING TO USE SERVER SIDE CONTROL OBJECT

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a framework to enable a developer to dynamically create and process a web page with a minimum program.

SOLUTION: Data joining of the property of a control object to be created in control object hierarchy is performed to the property of a server side data source and hierarchical data joining relation is established between the properties of the control object and the data source. Template declaration is used to regulate the structure of joined container object corresponding to a data object of the data source. A repetition control object judges the number of data objects to be increased according to the number of joined container objects. Types of a single data joining include (1) unidirectional



data joining from the data source to the control object, (2) unidirectional data joining from the control object to the data source and (3) bi-directional data joining between the control object and the data source.

LEGAL STATUS

[Date of request for examination] 04.12.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the server computer connected to the client computer system It is the approach of carrying out data coupling of the property of a lower layer server side control object to the property of the data set in the server side data table which has one or more data sets. Said lower layer server side control object supports the client side user interface, and repeats and refers to each data set of said server side data table. The process which creates the joint container server side control object corresponding to each data set, The process which relates each joint container server side control object with one of the data sets of said server side data table, It is in the lower layer of the joint container server side control object corresponding to a given data set as a property of each data set. And the process which creates the lower layer server side control object related with the property of said given data set, The approach including the process which makes data-coupling relation establish between the property of said lower layer server side control object, and one property in the data set of said server side data table of carrying out data coupling.

[Claim 2] The process which loads said server side data table from a server side database, The process which relates a repeat server side control object with said server side data table, The process which creates said repeat server side control object in the server side control object hierarchy each of whose joint container server side control object is a child object of said repeat server side control object, The approach according to claim 1 of carrying out data coupling which includes further the process which stores said server side data table as a property of said repeat server side control object.

[Claim 3] The approach including the process which sets up the data source property of said repeat server side control object in the process which associates said repeat server side control object in order to refer to said server side data table according to claim 2 of carrying out data coupling.

[Claim 4] The approach including the process at which each joint container server side control object creates each joint container server side control element related with the given data set of said server side data table as a child of said repeat server side control object in said repeat process according to claim 2 of carrying out data coupling.

[Claim 5] The approach including the process which creates the joint object containing the control identifier of said lower layer server side control object, the 1st property identifier of the property of said lower layer server side control object, the joint container identifier of said joint container server side control object, and the 2nd property identifier of the property of the data set of said server side data table in said process made to establish according to claim 1 of carrying out data coupling.

[Claim 6] The approach according to claim 1 of carrying out data coupling which includes further the process which transmits the data stored as a property of said lower layer control object based on said data-coupling relation since it stored as a property of the data set of said server side data table.

[Claim 7] The approach according to claim 1 of carrying out data coupling which includes further the process which receives the data stored as a property of said server side data table based on said data-coupling relation since it stored in the property of said lower layer control object.

[Claim 8] In the server computer connected to the client computer system Shape is taken by the

computer system by the subcarrier. The property of a lower layer server side control object It is the computer data signal which codes the computer program which carries out executive operation of the computer process data which carries out data coupling to the property of the data set in the server side data table which has one or more data sets. Said lower layer server side control object supports the client side user interface element. Said computer process Each data set of said server side data table is repeated and referred to. The process which creates the joint container server side control object corresponding to each data set, The process which relates each joint container server side control object with one of the data sets of said server side data table, It is in the lower layer of the joint container server side control object corresponding to a given data set as a property of each data set. And the process which creates the lower layer server side control object related with the property of said given data set, A computer data signal including the process which establishes data-coupling relation between the property of said lower layer server side control object, and one property in the data set of said server side data table.

[Claim 9] In the server computer connected to the client computer system Reading [computer system] is possible. The property of a lower layer server side control object It is the computer program storage which is coding the computer program which performs the computer process data which carries out data coupling to the property of the data set in the server side data table which has one or more data sets. Said lower layer server side control object supports the client side user interface element. Said computer process Each data set of said server side data table is repeated and referred to. The process which creates the joint container server side control object corresponding to each data set, each joint container server side control object -- the data set of said server side data table -- with the process related with one It is in the lower layer of the joint container server side control object corresponding to a given data set as a property of each data set. And the process which creates the lower layer server side control object related with the property of said given data set, A computer program storage including the process which establishes data-coupling relation between the property of said lower layer server side control object, and one property of the data set of said server side data table.

[Claim 10] The property of a lower layer server side control object It is the computer program product which codes the computer program which performs the computer process for carrying out data coupling to the property of the server side data array which has one or more data objects on a computer system. Said child server side control object corresponds to a client side user interface element. Said computer process The process which loads said server side database array from a server side database, The process which relates a repeat server side control object with said server side data table, The process which creates said repeat server side control object in a server side control object hierarchy, The process which stores said server side data array as a property of said repeat server side control object, The process which is the child of said repeat server side control object, and creates one of said the data objects, and a corresponding joint container server side control object, The process which relates said joint container server side control object with one of the data objects of said server side data array, The creation process which is a process which creates a lower layer server side control object for each property of each data object, and has each lower layer server side control object in the lower layer of said joint container server side control object, A computer program product including the process which establishes data-coupling relation between the property of said lower layer server side control object, and the property of said server side data array.

[Claim 11] A computer program product including the process which sets up the data source property of said repeat server side control object in the process which associates said repeat server side control object in order to refer to said server side data array according to claim 10.

[Claim 12] A computer program product including the process which creates each joint container server side control element related with the given data object of said server side data array as a child of said repeat server side control object in said repeat process which associates said repeat server side control object according to claim 11.

[Claim 13] A computer program product including the process which creates the joint relation object containing the control identifier of said lower layer server side control object, the property identifier of the property of said lower layer server side control object, the joint container identifier of said joint

container server side control object, and the property identifier of the property of the data object of said server side data array in said process to establish according to claim 10.

[Claim 14] The computer program product according to claim 10 which includes further the process which transmits the data stored as a property of said lower layer control object based on said data-coupling relation since it stored as a property of the data object of said server side data array.

[Claim 15] The computer program product according to claim 10 which includes further the process which receives the data stored based on said data-coupling relation as a property of the data object of said server side data array for storing in the property of said lower layer control object.

[Claim 16] It is the server which performs server side data coupling using the hierarchy of the server side control object corresponding to a client side user interface element. The server side data array in which each data object contains the property and which has one or more data objects, The repeat server side control object which is in a server side control object hierarchy, and was related with said server side data array, One or more joint container server side control objects repeatedly created by said repeat server side control object based on many data objects in said server side data array, Each lower layer server side control is created in said server side data array as a child of said combined-control server side control object of a given data object. One or more lower layer server side control objects corresponding to the property of each data object of said server side data array, A server including the data-coupling relational structure which describes the data-coupling relation between the property of said lower layer control object, and the property of the data object of said server side data array.

[Claim 17] The server according to claim 16 which contains further the push module which stores the data from the property of said lower layer control object in the property of the data object of said server side data array based on said data-coupling relation.

[Claim 18] The server according to claim 16 which contains further the push module which stores the data from the property of the data object of said server side data array in the property of said lower layer control object based on said data-coupling relation.

[Claim 19] The server according to claim 16 which contains further a part of server side data storage section loaded to said server side data array.

[Claim 20] The server according to claim 16 which contains further a part of server side data storage section where said server side data array is saved.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] Generally especially this invention relates to the server side control object which combines the property of a server side data table in the two directions about a web server framework.

[0002]

[Description of the Prior Art] A typical web browser receives data from the web server which specifies the appearance of a web page and basic actuation which are displayed in a client system. As a typical procedure, a user specifies the uniform resource (resource) locator (henceforth "URL") which is the global address of the resource on World Wide Web, and a desired website is accessed. Generally, the vocabulary "a resource" is the data which can be accessed by the program, or a routine. An example of URL is "HYPERLINK "http://www.microsoft.com/ms.htm" http://www.microsoft.com/ms.htm." The 1st part of this example of URL shows the given protocol (for example, "http") used for a communication link. The 2nd part specifies the domain name (for example, "HYPERLINK "http://www.microsoft.com" www.microsoft.com") which shows the location of a resource. The 3rd part specifies the resource in a domain (for example, file called "ms.htm"). This is followed and they are a browser and HYPERLINK "http://www.microsoft.com" www.microsoft.com. The HTTP (hypertext transport protocol) request relevant to the example of URL for taking out the data relevant to the ms.htm file in a domain is generated. The web server which is acting as the host of the www.microsoft.com site returns reception, the demanded web page, or a resource to a client system by the HTTP response, and displays a HTTP request on a browser.

[0003] The "ms.htm" file of the above-mentioned example contains the static HTML (HyperText Markup Language) code. HTML is a plain text authoring language used for the document (for example, web page) creation on World Wide Web. Since it is such, an HTML file can be taken out from a web server, can be displayed on a browser as a web page, and can offer the rich graphical experience which it comes to expect while the user is looking at the information from the Internet. A developer specifies the text and list which are displayed on a browser and by which formatting was carried out, form, a table, a hypertext link, an in-line image, voice, and background graphics using HTML. However, an HTML file is a static file which is not supporting dynamic generation of web page contents in essence.

[0004] When displaying dynamic contents, such as change of a stock price, and traffic information, the server side application program for treating a usually more complicated client-server dialogue is developed. A server side application program processes a HTTP request, and generates the suitable HTML code for transmission of the HTTP response to a client. An example of a HTTP request may contain parameters, such as data from the data or the web based form in an enquiry string. Since it is such, a server side application program can process this parameter, and can generate the HTML code dynamically by the HTTP response to a client. The document containing the suitable HTML code may be generated dynamically, using the sequence of one or more text write-in processings to a memory device by which formatting was carried out as an example of a server side application program. Then, it is transmitted to a client system by the HTTP response, and the obtained document is displayed on a

browser as a web page there.

[0005] Development of a server side application program is a complicated activity as which it is required it is not only well versed in the usual HTML coding used for a web page design, but that it is well versed in pro GURAMUBE six including one or more programming language (for example, C++, Perl, Visual Basic, or Jscript). However, a web page architect is graphic designer or an editor in many cases, and may be inexperienced in a program. Furthermore, if complicated web page development is simplified, the rate of development of new web contents can be gathered by any developers. Generally, a great effort is required and development of a custom-made server side application program is also actually like [which a developer regards as not wanting to often try it]. Therefore, it is desirable to offer the development framework to which a developer can create and process a web page dynamically by the minimum program.

[0006]

[Problem(s) to be Solved by the Invention] One means which makes min the requirements for a program of dynamic web page generation is an active server page (ASP) framework offered by Microsoft Corp. An ASP resource processes typically the HTTP request which specifies an ASP resource as a desired resource, after that, generates the HTML code as a result of the HTTP response to a client, for example, contains Visual Basic or Jscript. Furthermore, in order to mitigate a given application program effort, or the ASP resource was developed beforehand, refer to a third party's client side library component (for example, client side ACTIVEX control) for it. However, in a current server side application framework, programming required to manage dynamically client side user interface elements (for example, a text box, a list box, a carbon button, a hyperlink, an image, voice, etc.) within server side application needs a still advanced programming technique and a considerable effort. An unsolved technical problem is about processing a user interface element encapsulating programming demanded appropriately so that a web page developer can focus on other aspects of a web page.

[0007] A certain client side user interface element contains the data related with the server side data storage section like a database. An example of such an element is a client side table which displays the list of a product and prices from a server side product database. Moreover, the text box into which a consumer can input the shipping address which should be stored in a server side customer database as another example for next utilization is also included. The application program was written by the application developer, in order that a client side user interface might be written in a page by the page developer and might perform data communication between a page and a server side database by him conventionally. The problem is the point that both page development and application program development also requires a great development effort, and needs a database structure and the deep information about a certain amount of complicated programming (minding scree PUTINGU of a client side, or application programming by the side of a server).

[0008]

[Means for Solving the Problem] According to this invention, **** and other technical problems are solved by offering the hierarchy data which combine one or more server side control objects and the one or more server side data storage sections. Hierarchy data coupling is supported by creating a joint container automatically to each data object in a data array. Each joint container offers access to the public property containing the data object of the server side data storage section associated to the child. By doing in this way, the property of a server side control object may be combined with the property of the server side data storage section through the joint container. Data-coupling relation is not limited to this, although it is establishable from the property of a server side control object in one direction to the property of the server side data storage section, or its hard flow. Although data-coupling relation is also establishable on the 2 direction targets between the property of a server side control object, and the property of the server side data storage section, it is not limited to this again.

[0009] In the means concerning a certain this invention, the approach of combining data with the property of the data set (for example, table line) of a server side data table which has one or more data sets for the property of a lower layer server side control object is offered in the server computer linked to a client computer system. This lower layer server side control object corresponds to a client side user

interface element. A repeat control object is repeatedly referred to for every data set of a server side data table, and creates the joint container server side control object corresponding to each data set. Each joint container server side control object is related with the data set of a server side data table. A lower layer server side control object is created to the property of each data set. A given lower layer server side control object is in the lower layer of the joint container server side control object corresponding to a given data set, and is related with the property of a given data set. Data-coupling relation is established between one of the property of a lower layer server side control object, and the data sets of a server side data table.

[0010] In other means concerning this invention, a product is offered as a computer program product. In an operation gestalt with a computer program product, according to a computer system, reading is possible and the computer program storage which codes the computer program for performing the computer process which carries out data coupling of the property of a lower layer server side control object to the property of the server side data array which has one or more data objects is offered. As operation gestalt of a computer program product another again, it can also provide in the computer data signal which is materialized by the computer system by the subcarrier and codes a computer program. A child server side control object corresponds to a client side user interface element. A server side data array is loaded from a server side database. A repeat server side control object is related with a server side data table, and is created in a server side control object hierarchy. A server side data array is stored as a property of a repeat server side control object. One of the data objects and a corresponding joint container server side control object are created as a child of a repeat server side control object. A joint container server side control object is related with one of the data objects of a server side data array. A lower layer server side control object is created to each property of each data object. Each lower layer server side control object is in the lower layer of a joint container server side control object. Data-coupling relation is established between the property of a lower layer server side control object, and the property of a server side data array.

[0011] In another means, the server which performs server side data coupling using the hierarchy containing the server side data array which has one or more data objects of the server side control object corresponding to a client side user interface element is also offered. Each data object may also contain a property. The repeat server side control object in a server side control object hierarchy is related with the server side data array. One or more joint container server side control objects are repeatedly created by the repeat server side control object based on the data object of a large number in a server side data array. One or more lower layer server side control objects correspond to the property of each data object of a server side data array. Each lower layer server side control is created by the lower layer of the combined-control server side control object of the given data object in a server side data array. Data-coupling relational structure describes the data-coupling relation between the property of a lower layer control object, and the property of the data object of a server side data array.

[0012]

[Embodiment of the Invention] In an operation gestalt with this invention, on a web server, web page contents are generated dynamically and displayed by the client. A client can become any browsers which support for example, the criterion HTML or other authoring languages. The client in a web server communicates through a network by using a HTTP request, a HTTP response, etc. Thus, although a web server is the HTML code gestalt, it generates web page contents and transmits the contents to a client. Thereby, a web page can be displayed on a browser. The server side control object which can respond to each user interface element and logic target of a web page is created on a web server, and processes and carries out the rendering of the web page contents. A server side control object is declared with dynamic contents resources, such as an ASP+ resource processed by the page factory which creates the hierarchy of a server side control object. The control object in a hierarchy collaborates and processes the request received from the client, subsequently, it generates result web page contents, transmits to a client, and terminates the control object in a hierarchy.

[0013] In an operation gestalt with this invention, a page object can be illustrated as a control object hierarchy's top level. In the page object which is also a control object, each child control object spreads

to the hierarchy of a large number level including one or more child control objects including one or more child control objects in itself typically. A page object and a lower layer control object perform a processing sequence, and process and generate the web contents corresponding to a client side user interface element.

[0014] One of the processings in this sequence includes the data-coupling processing which controls the communication link with one or more server side control objects and server side data storage section. In an operation gestalt with this invention, the server side data storage section is a database which stores the data about the user interface element of a web page (see the data storage section 131 of drawing 1). Although not limited especially, other non-database mold data storage sections containing configuration registration, a data file, and a data stream are contained within the limits of this invention. By declaring a server side control object and the data-coupling relation of relation in a dynamic contents resource, a page developer can constitute an one direction and/or the control object hierarchy who has the 2 direction data-coupling relation to the server side data array relevant to the server side data storage section.

[0015] Drawing 1 shows the web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention. A client 100 performs the browser 102 which displays the web page 104 on the indicating equipment of a client 100. A client 100 may also include the client computer system which has a display like a video monitor. The "INTERNET EXPLORER" browser currently sold by Microsoft Corp. is an example of the browser 102 in an operation gestalt with this invention. As an example of other browsers, although there are "NETSCAPE NAVIGATOR", "MOSAIC", etc., it does not restrict to this. The table list 140 which has four list line elements is built into the illustrated web page 104. The list line element 142 is a child user interface element which the table list element 140 contains. The list line element 142 is defined by template declaration (see the drawing 6 A as an example of template declaration), and has two trains containing the label elements 144 (for example, OrderID is expressed) and 146 (for example, the amount of an order product is expressed), and one train containing the table list element 148. Elements 144, 146, and 148 are declared as a child user interface element contained with the list line element 142 (see the drawing 6 A as an example of a label and table list declaration). The addition list line elements 150, 152, and 154 are also shown in the web page 104.

[0016] According to declaration of drawing 6 A, data coupling of the server side control object corresponding to the user interface element of a web page 104 is carried out to a server side [two] data array (for example, the server side data tables MyOrders and MyItems relevant to server side database access through a MyOrderSystem object). Generally, a server side data array contains an index data set (for example, a data line or a data object) in a control object hierarchy in relation to the server side data storage section (for example, it is loaded from the storing section and stored in/or the storing section). Such a data set has the property with which the data value of a certain table data line can be expressed, for example. The column 148 expresses the sub table list element which has two, two trains, i.e., the train of the line index from the Items data table by which (1) association was carried out, and train [of three list line elements which contain a label (for example, subject name) from (2) Items data table] **.

[0017] a browser 102 -- the HTTP response 112 -- setting -- a web server 116 to the HTML code -- reception -- things are made and the web page described by the HTML code is displayed. Although HTML is explained with reference to a certain operation gestalt, especially, it is not restricted and it is thought that it is the markup language of SGML (Standard Generalized Markup Language) and the XML base, and other authoring languages containing WML (Wireless Markup Language) designed so that the content and user interfaces of narrow-band radio equipment, such as a pocket bell (trademark) and a cellular phone, might be specified are within the limits of this invention. Furthermore, although the criterion HTML 3.2 is mainly indicated in this description, any versions of HTML may be contained within the limits of this invention.

[0018] The communication link with a client 100 and a web server 116 can be performed using a series of processings of the HTTP request 114 and the HTTP response 112. Although HTTP is explained with reference to a certain operation gestalt, it is not restricted especially and it is thought that other transport

protocols including S-HTTP are within the limits of this invention. In a web server 116, the HTTP pipeline module 118 receives the HTTP request 114, analyzes URL, and calls the suitable handler 120 which processes a request. The web server 116 is equipped with two or more handlers 120 treating the resource of a different type in the operation gestalt with this invention.

[0019] For example, when URL specifies a static contents resource 122 like an HTML file, a handler 120 accesses the static contents resource 122, and transmits the static contents resource 122 to a client 100 by the HTTP response 112 through the HTTP pipeline 118. Or in an operation gestalt with this invention, when URL specifies a dynamic contents resource 124 like an ASP+ resource, a handler 120 accesses the dynamic contents resource 124, processes the contents of the dynamic contents resource 124, and generates the HTML code the result for web page 104. In an operation gestalt with this invention, the result HTML code contains standard HTML3.2 code. Generally, a dynamic contents resource is the server side declaration data storage section (for example, ASP+ resource) which can use the authoring language which describes the web page which should be displayed on a client for generating dynamically. And the HTML code for web pages passes the HTTP pipeline 118, and is sent to a client 100 by the HTTP response 112.

[0020] During this processing, the handler 120 was beforehand developed again, in order to make a development effort simplification, or it can access the library of the 3rd person code. One of such the libraries is the server side class control library 126, and a handler 120 can illustrate from here the server side control object which generates HTML data as a result of processing a user interface element and displaying on a web page. in an operation gestalt with this invention, one or more server side control objects are visible on the web page described by the dynamic contents file 124 -- it hides-like and maps to one or more user interface elements.

[0021] On the other hand, the 2nd library is a client side control class library 128 like the library containing the "ACTIVEEX" component from Microsoft Corp. "ACTIVEEX" control is a COM (component object model) object which follows a fixed criterion in the method of a dialogue with a client and other components. Client side "ACTIVEEX" control is the component of the COM base as for which downloads to a client automatically and executive operation may be carried out to it by the web browser of a client. A server side ACTIVEEX component (not shown) is a component of the COM base which may be performed on a server, in order to achieve the various server side functions in which the server side functionality of stock price retrieval application or a database component is offered. ACTIVEEX is indicated by the detail by "understanding of ACTIVEEX and OLE" (David Chappelle, the Microsoft press, 1996).

[0022] In contrast with "ACTIVEEX" control, the server side control object of the operation gestalt of this invention specified as the dynamic contents resource 124 corresponds to the user interface element built into the web page on a client logically. A server side control object can generate the effective HTML code which may contain the locator which carries out the reference of the given client side "ACTIVEEX" control to for example, a HTML tag again. When the browser has already had the code for client side "ACTIVEEX" control in the storing system, executive operation of the "ACTIVEEX" control is carried out within the web page on a client. If it becomes right [that], a browser will download the code for "ACTIVEEX" control from the resource specified by the locator, and will carry out executive operation of the "ACTIVEEX" control within the web page on a client. The server side control object in the operation gestalt of this invention can start an event again to the server side "ACTIVEEX" control used for performing stock price retrieval application on a server.

[0023] A handler 120 accesses one or more non-user interface server components 130 which carry out executive operation on a web server 116 or another accessible web server again. A non-user interface server component 130 like stock price retrieval application or a database component is referred to in the dynamic contents resource 124 processed by the handler 120, or is related with it. The server side event started by the control object declared with the dynamic contents resource 124 may be processed in server side code which calls the suitable method of the non-user interface component 130.

Consequently, the processing offered by the server side control object can simplify programming of non-user interface server component stereo-NENTO 130 by encapsulating processing and generation of

a web page of a user interface element, and, thereby, the developer of the non-user interface server component 130 can be concentrated on development of the function of the application proper instead of a user interface problem.

[0024] In the operation gestalt with this invention, data coupling of the property of a server side control object is carried out to the property in the server side data storage section 131. For example, data coupling enables a page developer to relate the property of the control object in a server side control object hierarchy with the data item in a server side database. Each control object updates that data-coupling property during this processing by taking out the property value related from the data object in a database (it is called one direction data coupling from the server side data storage section). Or each control object updates the property of the server side data storage section again using a control object inner property value (it is called one direction data coupling to the server side data storage section). The 2 direction data coupling can be offered between a server side control object and the server side data storage section combining these two types of one direction data coupling.

[0025] Drawing 2 shows the flow chart of processing of the client side user interface element using the server side control object in an operation gestalt with this invention, and generation processing. In the transmitting processing 200, a client transmits a HTTP request to a server. A HTTP request contains URL which specifies resources, such as an ASP+ resource. In reception 202, a server calls the suitable handler which receives a HTTP request and processes the specified resource. Reading appearance of the ASP+ resource is carried out by the syntax-analysis processing 203. The creation processing 204 generates a server side control object hierarchy based on the content of the specified dynamic contents resource (for example, ASP+ resource).

[0026] In processing 206, a control object hierarchy's server side control object performs postback event handling, postback data handling, status management, and one or more processings in data coupling. From a client, the postback event and data (collecting "postback input") from a user interface element are sent to a server, and are processed. Especially a postback event is not restricted and may also contain the "data change" event from the client side text box element sent to a "mouse click" event or a server from a client side carbon button element. Although postback data are not restricted to this, they may also contain the text inputted into the index of the item chosen from the text box element or the drop down box by the user.

[0027] In processing 208, each server side control object in a hierarchy is called in order to generate data like the HTML code for the display by the web page of a client side user interface element (or rendering). Although the vocabulary "a rendering" may mean the processing which displays graphics on a user interface, in this description, the "rendering" also means generation processing of the authoring language data which may be interpreted by client application like the browser for a display and a client side function. More detailed explanation of processing 206 and the rendering processing 208 is given in connection with drawing 6. In a certain operation gestalt, the call of the render() method in each control object is performed using a tree traversal sequence. That is, the call of the render() method of a page object becomes the repeat traverse covering the suitable server side control object in a hierarchy. As an option which calls the render() method of a suitable control object, approaches, such as event signaling or the object registration technique, may be used. A parenthesis specifies the "render()" label in which the method of comparing with a data value is shown.

[0028] In an operation gestalt with this invention, actual creation of each server side control object may delay until a server side control object is accessed in processings 206 or 208 (handling of a postback input, loading of a condition, rendering of a control object to the HTML code, etc.). Server processing is optimized when a server side control object is not accessed for a given request, and creation of a control object eliminates delay and unnecessary control object creation processing.

[0029] In the transmitting processing 210, the HTML code is transmitted to a client by the HTTP response. In reception 214, a client receives the HTML code relevant to the new web page which should be displayed. In display processing 216, a client system incorporates the user interface element of a new page according to the HTML code received from the HTTP response (for example, it displays).
however, a user interface element -- incorporating -- he should also understand that you may include

non-display processing of control of offer of voice or a tactile-sense output, read-out to memory and writing, and script processing etc. A server side control object hierarchy is ended in a post process 212. In an operation gestalt with this invention, the server side control object in a hierarchy answers the HTTP request which refers to the associated ASP+ resource, is created, and after the rendering of authoring language data (for example, HTML data) finishes, it is destroyed. In another operation gestalt, processing 212 may be performed before processing 210 after processing 208.

[0030] Drawing 3 shows an example of the module in the web server used in an operation gestalt with this invention. A web server 300 accepts the HTTP request 302 in the HTTP pipeline 304. The HTTP pipeline 304 may also contain various modules, such as logging of web page statistics, user collating, a right of user access, and a module for the formation of an output cache of a web page. Eventually, each input HTTP request 302 received by the web server 300 is processed by the specific instance of an IHTTP handler class (it illustrates as a handler 306). A handler 306 analyzes a URL request and calls a suitable handler factory (for example, page factory module 308).

[0031] In drawing 3, the page factory module 308 related with the ASP+ resource 310 is called, and instantiation and configuration of the object of the ASP+ resource 310 are handled. In a certain operation gestalt, an ASP+ resource may be recognized by specifying a specific suffix (or a file extension part like ".aspx") as a file. If the request to a given ASP+ resource is first received by the page factory module 308, the page factory module 308 will search a file system, and will obtain a suitable file (for example, .aspx file 310). This file may also contain the text (for example, authoring language data) which may be behind interpreted or accessed by the server which processes a request, or the data (for example, cutting tool code data or coded data) in another format. When a physical file exists, the page factory module 308 reads a file to an aperture, and reads the file to memory. When a file is not found, the page factory module 308 returns the suitable error message "a file is not found."

[0032] After reading the ASP+ resource 310 to memory, the page factory module 308 processes a file content, and builds the data models (for example, a list [of script blocks], directive, static text area, and hierarchy server side control object, a server side control property, etc.) of a page. A data model is used for generating the source list of a new object class like the COM+ (Component Object Model+) class to which the class of the page base is made to extend. The page base class contains the code which defines the structure, property, and function of a basic page object. In an operation gestalt with this invention, it is compiled dynamically [a source list] subsequently to intermediate language, and is the instruction of a proper to a platform behind. s (for example, X86, Alpha, etc.) It is compiled by timely (Just-In-Time). Intermediate language is COM+ IL. A code and Java A cutting tool code and Modula 3 A code and SmallTalk A general purpose or custom-made orientation linguistic code of a code, the Visual Basic code, etc. may also be included. In another operation gestalt, intermediate-language processing may be excluded and the instruction of a proper may be directly generated from a source list or a source file (for example, ASP+ resource 310). The control class library 312 may be accessed with the page factory module 308, in order to obtain the server side control class which is used for a control object hierarchy's generation and which was defined beforehand.

[0033] The page factory module 308 outputs the page object 314 which is a server side control object equivalent to the web page 104 of drawing 1. The page object 314 and its child object are the control object hierarchy's 316 examples. It can think according to this invention, and like the custom-made control object, the example of other control objects is not restricted especially, either and also contains the object corresponding to HTML control of a table 1. The page object 314 corresponds to the web page 104 of drawing 1 logically, and relates to other control objects and hierarchy targets on a server. In a certain operation gestalt, a page object is a container object which contains the child control object hierarchical. The hierarchical relationship of other gestalten, such as a dependency, can be used with another operation gestalt. In a complicated control object hierarchy, one child object may be a container object of other child objects rather than it has the child object of a large number level (for example, the table list 140 is the container of the list line elements 142, 150, 152, and 154 and the child element of those). These server side control objects collaborate, handle the input from the HTTP request 302, manage the condition of a server side control object, perform data coupling, and carry out the rendering

of the authoring language data (for example, HTML) for a web page as a result of a client. The result HTML code is an output from the control object hierarchy 316, and is transmitted to a client by the HTTP response 324.

[0034] It sets in the above-mentioned operation gestalt, the control object hierarchy's 316 control object is created and performed on a server 300, and each server side control object corresponds to the user interface element and logic target on a client. A server side control object collaborates, handles the postback input from the HTTP request 302 again, manages the condition of a server side control object, performs data coupling with a server side database, and generates the authoring language data (for example, the HTML code) used for the display of a web page as a result of a client. Result authoring language data are generated from the server side control object hierarchy 316 (namely, rendering), and are transmitted to a client by the HTTP response 324. For example, the result HTML code can embody any effective HTML configurations, and in addition to this, control of an ACTIVEX type, a JAVA (trademark) applet, a script, and when it is processed by the browser, refer to any web resources which produce client side user interface elements (for example, a control carbon button, a text box, etc.) for it.

[0035] By declaration made with the ASP+ resource 310, a server side control object can access one or more non-user interface server components 330 for a dialogue with the non-user interface server component 330 and a client side user interface element. For example, a postback input can be answered and a server side control object can start a server side event to the non-user interface server component registered into those events. Thus, the non-user interface server component 330 minds a user interface element for a dialogue with a user, and it can perform it, without programming a code required displaying and processing these elements.

[0036] Data coupling of the property of the one or more server side data storage sections 331 can be carried out to the property of a server side control object. Detailed explanation about a server side control object hierarchy's example constituted so that one gestalt of data coupling might be supported is given in relation to [drawing 7](#). This data-coupling relation is declared in an ASP+ resource, it compiles a server side control object, and while carrying out run time activation, it is established.

[0037] In short, the operation gestalt of this invention contains on a server creation and the server side control object by which executive operation is carried out, in order to generate the HTML code transmitted to a client. The HTML code can embody any effective HTML configurations, and is control of an ACTIVEX type and JAVA, for example. Any web resources of the others which generate the user interface carbon button and other user interface elements in an applet, a script, and a client can be referred to. The user in a client can have a dialog with these user interface elements that correspond to a server side control object logically, and can return a request to a server. A server side control object is re-created on a server, and it processes the data, the event, and other properties of a user interface element so that the HTML code of the next round which should transmit to a client as a response may be generated.

[0038] If [drawing 4](#) is referred to, an example of the computer system of the operation gestalt of this invention contains the general purpose computer equipment of the gestalt of the conventional computer system 400 containing the system bus 406 which connects the various system components containing the processor unit 402, a system memory 404, and a system memory 404 to the processor unit 400. System buses 406 may be any of the bus structure of some types containing the peripheral bus and local bus which use a memory bus or a memory controller, and various bus architectures. The system memory contains the memory (ROM) 408 only for playbacks, and random access memory (RAM) 410. The unformatted input / output system 412 (BIOS) containing the basic routine which helps a transfer of the information between the elements within a computer system 400 are stored in ROM408.

[0039] The computer system 400 contains the optical disk drive 418 which performs read-out and the writing of a removable optical disk 419 like the magnetic disk drive 414 and CD ROM which perform further read-out and the writing of a hard disk drive 412 and the removable magnetic disk 416 which perform read-out and the writing of a hard disk, DVD, or other optical media. The hard disk drive 412, the magnetic disk drive 414, and the optical disk drive 418 are connected to the system bus 406 by the hard disk drive interface 420, the magnetic disk drive interface 422, and the optical disk drive interface

424, respectively. A drive and its medium which can be related computer read offer the instruction of a computer system 400 which can be computer read, DS, a program, and the non-volatile storage section of other data.

[0040] Although the hard disk, the removable magnetic disk 416, and the removable optical disk 419 are used for this description in the above-mentioned environmental example of a publication, the medium type [other] in which data storage is possible and which can be computer read can be used for the above-mentioned example of a system. The medium of other types of these which can be used for the above-mentioned example of operating environment which can be computer read has for example, a magnetic cassette, flash memory card, a digital videodisc, a BERUNUI (Bernoulli) cartridge, random access memory (RAM), the memory (ROM) only for playbacks, etc.

[0041] Many program modules may be stored in a hard disk, a magnetic disk 416, an optical disk 419, and ROM408 or RAM410, and these contain the application program 428, other program modules 430, and the program data 432 of 426 or 1 or more operating systems. A user can input a command and information into a computer system 400 with input units, such as a keyboard 434 and a mouse 436, or other pointing equipments. As other input devices, there are a microphone, a joystick, a gamepad, a satellite dish, a scanner, etc., for example. These and other input devices are connected to the processor 402 through the serial port interface 440 connected to the system bus 406 in many cases. However, these input devices may be connected by other interfaces, such as a parallel port, a game port, or Universal Serial Bus (USB), again. The monitor 442 or the indicating equipment of other types is also connected with the system bus 406 through the interface of a video adapter 444 etc. In addition to a monitor 442, typically, a computer system contains other circumference output units (not shown), such as a loudspeaker and a printer.

[0042] A computer system 400 can operate in the environment using the logical connection to one or more remote computers like a remote computer 446 connected by network. a remote computer 446 -- a computer system, a server, a router, Network PC, and a peer (peer) -- it may be equipment or other common network nodes, and there are many elements typically mentioned above in connection with a computer system 400, or all are included. Network connection contains Local Area Network (LAN) 448 and Wide Area Network (WAN) 450. Such a network environment is not new in office, an enterprise magnitude computer network, intranet, and the Internet.

[0043] When using by the LAN network environment, a computer system 400 is connected to a local network 448 through a network interface or an adapter 452. When using by the WAN network environment, a computer system 400 includes the modem 454 or other means for establishing the communication link by Wide Area Network 450 like the Internet typically. Built-in or external any is sufficient as a modem 454, and it is connected with the system bus 406 through the serial port interface 440. In the environment connected by network, the program module described in relation to the computer system 400 or its part may be memorized by remote memory storage. The illustrated network connection is an example and can use other means for the communication link establishment between computers.

[0044] In the operation gestalt of this invention, a computer 400 expresses a web server and a processor 402 carries out executive operation of the page factory module on the ASP+ resource memorized by at least one of storages 416, 412, 414, 418, and 419 or memory 404. A HTTP response and a request communicate by LAN448 connected to the client computer 446.

[0045] Drawing 5 is a process flowchart showing server side processing of the page object in an operation gestalt with this invention, and other control objects. In processing 500, the page object construction section is called with the page factory module 308 (refer to drawing 3). Consequently, a page object (see the page object 314 in drawing 3) is created so that it may correspond to the web page user interface element on a client logically. In processing 502, a page factory module calls the ProcessRequest member function of the page object which begins to process [gradual] the HTTP request received from the client. In the 1st phase of 1 operation gestalt of this invention, server side creation processing (not shown) creates the lower layer server side control object contained in the control object hierarchy of a page object. That is, in order that the construction section of a child control

object may create a control object among the duration of HTTP request processing, call appearance is carried out repeatedly.

[0046] However, in another operation gestalt, creation of a child control object is delayable until a control object is needed for a given processing step (for example, rendering of the HTML code of the user interface element with which handling of a postback event, handling of postback data, loading of a view state and preservation, and data coupling analyze or correspond). Since the operation gestalt of the latter "which delays control object creation" can reduce utilization of unnecessary CPU and memory, it is the optimal. For example, there is a case where it will be called creation of the web page from which the user input event received from the client completely differs. In this case, a control object hierarchy is terminated promptly, and only in order to process the event which will illustrate a new different control object hierarchy of a new page, it is not necessary to illustrate the whole control object hierarchy of a former page.

[0047] The server call of the ProcessRequest method of a page object can be answered, and executive operation of the processings 504-520 can be carried out by the page object and each lower layer control object according to the data of a given HTTP request etc. In an operation gestalt with this invention, processings 504-520 are performed to each object of each in the sequence of drawing 5. However, depending on a HTTP request, about given processing of another object, it may not be carried out in order or, as for the given processing to one object, processing may not be performed at all. For example, the 1st object performs the initialization processing 504 and its load processing 506, the postback data processing 508 is started and a lower layer control object performs the initialization processing 504 and the load processing 506 of itself by overdue control object creation after that. Although the sequence of processing by the page object and the lower layer control object is not restricted to this, it depends for it on whether the current condition of the property of the data in a HTTP request, a control object hierarchy's configuration, and a control object and control object creation were overdue.

[0048] The initialization processing 504 initializes a control object, after a control object is created by carrying out executive operation of the server side code of the arbitration about initialization in a dynamic contents resource. Thus, each server side control object may be customized by the specific server side function declared with the dynamic contents resource. In an operation gestalt with this invention, a dynamic contents code customizes or extends the base page control class declared by the page developer with the ASP+ resource on a server. Compile of an ASP+ resource contains the declared code in the suitable initial code (for example, Init() method of a page object and a lower layer control object). The initialization processing 504 carries out executive operation of this code, and customizes or extends a page base class and the base class of a lower layer control object.

[0049] In an operation gestalt with this invention, the status management of a server side control object is supported in the load processing 506 and the preservation processing 516 in which the condition structure which can be conveyed is used, in order to contain the non-statement model of a client/server system by returning a server side control object to a former condition. With a certain operation gestalt, although a condition goes and comes back to a server with another next condition structure [in the one or more HTML fields in which the HTTP request / response of a couple hid] which can be conveyed, it is considered that other condition structures which can be conveyed are within the limits of this invention.

[0050] In a series of processings by the given present request and present given response about a page between a client and a server, the condition of one or more control objects is recorded on the condition structure which can be conveyed by the preservation processing 516 after processing of a previous request. In an operation gestalt with this invention, the status information of the addition containing the control object identifier which enables a hierarchy or a server to associate a suitable control object and a given condition is also included in the condition structure which can be conveyed. In the next HTTP request, status information is returned to a server with the condition structure which can be conveyed. A server extracts status information from the received condition structure which can be conveyed, loads condition data to the suitable control object in a control object hierarchy, and returns each control object to the condition that it existed before the previous HTTP response. After the processing to the present

request, again, the condition of one or more server side control objects is recorded on the condition structure which can be conveyed by the preservation processing 516, and returns the condition structure which can be conveyed to a client by the next HTTP response by it.

[0051] Each server side control object is changed into the same condition as the condition before a previous HTTP request as a result of the load processing 506. For example, when a text box control object includes a property value equivalent to "JDoe" before a previous HTTP response, the load processing 506 returns the same control object to a previous condition by loading text string "JDoe" to the property value etc. Furthermore, whether the condition of a given object was memorized and recovered can also constitute.

[0052] When an operation gestalt with this invention is summarized, "it is saved" after the condition of one or more server side control objects processing. The saved status information is transmitted to a client by the response. A client returns the status information saved by the next response to a server side. A server loads status information to the newly illustrated server side control object hierarchy so that a hierarchy's condition may return to a former condition.

[0053] With another operation gestalt, status information can be held for some other accessible web locations by the server during round trip on a server of returning from a server to a client and a server. After a server receives a client request, this status information is taken out by the server and may be loaded to the suitable server side control object in a control object hierarchy.

[0054] In processing 508, the postback data received from the HTTP request are processed. Postback data may be contained in the pay load of the HTTP request by the key value and hierarchical display (for example, XML) which became a pair, or other data display like RDF (Resource Description Framework). Processing 508 analyzes the syntax of a pay load, and identifies the identifier of the meaning of a Target server side control object. When an identifier (for example, "page:tablelist1:listrow2:label1") is found and a Target server side control object exists in a control object hierarchy, corresponding postback data are passed to the control object. For example, reference of drawing 1 sends the identifier and associated data of a meaning relevant to the data item of the list line 142 and a train 144 to a web server 116 with the pay load of the HTTP request 114. Processing 508 analyzes the syntax of the pay load of the HTTP request 114, and acquires the identifier and its related value (namely, the HTML code of a child user interface element like the list line 142) of a meaning of the table list 140. And processing 508 passes a postback value to the target control object which analyzes the identifier of the meaning of the list line 142, identifies and traverses and processes a corresponding server side control object.

[0055] As the load processing 506 was explained, the property value of a server side control object may be returned to a former condition. If postback data are received, a server side control object will judge whether the property value of the point to which the passed postback value corresponds was changed. When it changes, logging of the change is carried out to the change list in which data change of a related control object is shown. After all postback data are processed within a control object hierarchy, a control object method is called and one or more postdata change events can be started to one or more non-user interface server components like stock price retrieval application started on a server. A postback data change event is an event which shows that for example, postback data changed the property of a server side control object. In an instantiation operation gestalt, such an event is sent to a system offer event queue, and can call the server side code registered so that an event might be processed. Subsequently, a server side code can call the method of a non-user interface server component. Thus, a server side non-user interface server component can answer the event in which the trigger was carried out by change of the data of a server side control object. The option including using an application offer event queue, polling, and processing interruption which performs an event can also be considered to be within the limits of this invention.

[0056] A postback event is handled in processing 510. A postback event communicates with the pay load of a HTTP request. Processing 510 analyzes the syntax of the specific event target (for example, with the operation gestalt with this invention, label attachment is carried out with "#EVENTTARGET") which identifies the server side control object to which the event is turned. Furthermore, processing 510

analyzes the syntax of it, when there is a discovered event argument, and it gives it to the server side control object which had the event argument (for example, label attachment is carried out with "#EVENTARGUMENT" with the operation gestalt with this invention) specified. A control object starts the event processed with the non-user interface server component relevant to a dynamic contents resource.

[0057] Processing 512 solves the data-coupling relation between a server side control object and one or more server side databases with an accessible server, and updates a control object property with a database value by this, and updates the database field with the value of/or a control object property. the property of a server side control object may be related with the property of a parent data-coupling container as shown in the table of a server side application database with an operation gestalt with this invention (or data coupling is carried out -- having). A page frame work piece can update the control object property which has the value of the parent data-coupling container property corresponding to between the data-coupling processings 512 and by which data coupling was carried out. Thus, the user interface element on the web page in the next response reflects the updated property value in accuracy. The control object property with which a user interface element corresponds is because it was automatically updated between the data-coupling processings 512. Similarly, a control object property may be updated by the parent data-coupling container field, and, thereby, updates a server side application database in the postback input from a server side control object again.

[0058] The data-coupling relation of a given server control object and a given given data source (for example, server side data table) is declared in an ASP+ resource (see the line 14 of drawing 6 A). In order to support the common hierarchical property of joint data in a web page, data-coupling relation can support the hierarchy of itself (see the explanation of drawing 7). In a control object hierarchy, data-coupling relation identifies the property of the data table with which the property of a data table and a server side control object is combined. Although data-coupling relation may be two directions again, this means that the property value of (1) server side data table may be loaded to the joint property of a server side control object, and the property value of (2) server side control object may be loaded to the joint property of a server side data table. All server side control objects reflect in accuracy the data-coupling property value appropriately updated from the server side data table after the data-coupling processing 512.

[0059] Processing 514 updates a large number by which executive operation may be carried out, before saving a control object condition and carrying out the rendering of the output. Processing 516 requires status information (namely, view state) from one or more control objects in a control object hierarchy, stores status information, and inserts it in the condition structure which is sent to a client with a HTTP response pay load and which can be conveyed. For example, a "grid" control object saves the present index page of the list of values so that a "grid" control object can return to this condition after the next HTTP request (namely, processing 506). As mentioned above, view state information expresses the condition of the control object hierarchy before the next action by the client. It will be used for changing a control object hierarchy into the condition of the point before client post back input process or data coupling if view state information returns.

[0060] The rendering processing 518 generates the suitable authoring language output (for example, HTML data) transmitted to a client by the HTTP response. A rendering is performed by top-down hierarchy tree WOKU and the embedded rendering code of all server side control objects. Processing 520 does the last clean-up activity (it is connection of a database in closing a file ****) of arbitration, and a control object hierarchy is terminated. Subsequently, processing performs return and processing 522 to 502, and a page object is ended by calling the destructive section there.

[0061] Drawing 6 A shows an example of a dynamic contents resource (for example, ASP+ resource) which declares a data-coupling control object to a server in the operation gestalt of this invention. A line 1 contains as a literal <html> which is the initiation tag of an HTML file. Up to lines 2-8, the data array of local "MyOrders" and "MyItems" expresses the code declaration block of a code which carries out a POPYU rate. Specifically, a line 2 declares the code by the side of a server as a server side script with a "runat=server" attribute and a value. This script makes an invalid load processing 506 of drawing 5 of a

page object. Lines 3-5 declare an accessible public object in the name tooth space of a server side control object. "MyOrderSystem" of the "OrderSystem" type which is a public object is declared, this is used with a line 7, and a line 3 takes out the data array declared as "MyOrders" of a public data array object in the line 4 from the server side data storage section. It is used also for "MyOrderSystem" of a public object taking out the data array declared as "MyItems" of the public data array object in a line 5 from the server side data storage section again in a line 8.

[0062] A line 11 contains the literal which is the initiation tag <body> of the text of an HTML file. A line 12 contains the literal HTML code which specifies header"H1" which displays on a web page. A line 13 includes declaration of a server side form control object. A line 14 declares the server side table list control object which has a data-coupling data source property equal to "Page.MyOrders." Therefore, a MyOrders data table is declared as a table list control object and its lower layer data source.

[0063] In-line template declaration of lines 15-24 declares one or more child control objects of the table list control object declared with the line 14. Template declaration is interpreted at the time of syntax analysis, repeats a suitable ITemplate interface instance and combines it dynamically to the property of the same name in a control object (for example, table list control object). A table list control object (namely, repeat control object) uses an ITemplate instance at the time of activation, and creates and initializes the instance of N from 0 with a suitable new template control class. The number of the created new instances is controllable by the number of data sets (for example, data line) with which a data source corresponds.

[0064] For example, lines 15-24 declare a template called "itemtemplate" to the list line control object of the table list control object declared in the line 14. The parameter of a template control object is declared in lines 16-22, and shows the list line control object containing an OrderID label control object, a Quantity label control object, and other table list control objects. The table list control object declared in the line 14 repeats and refers to the data set of a data table MyOrders, and creates a lower layer list line control object to each data set.

[0065] The list line control object corresponding to an itemtemplate template is developed by the control object developer as an IBindingContainer instance, therefore it is considered that it is a "joint container." Between the load processings 506 of drawing 5, the POPYU rate of the MyOrders data table is carried out by the code of a line 7 from MyOrderSystem. For example, the data offered by the MyOrdersSystem object are extracted from the server side database containing an order and product data. Then, a table list control object creates a list line control object as a joint container to each data set of a data table MyOrders.

[0066] The OrderID control object declared in the line 16 is combined with the OrderID property of the data set in a MyOrders data table. The hierarchy data-coupling relation of the OrderID label control object and list line control object which were defined by the PropertyBinding object of an OrderID label control object establishes that the Text property of the label control object declared in the line 20 is data combined with the data item property identified as "OrderID". Hierarchy data-coupling relation specifies that a data item property is the joint container of a control object, i.e., the property of a MyOrders data table, again. Although the property of the MyOrders data table which is combined and serves as an OrderID control object is a data item, any public properties of a server side data array (for example, a data table or a data object another type) are combinable with the property of a server side control object including a line index, data value size, a data value type, etc.

[0067] The Quantity control object declared in the line 17 is combined with the Quantity property of the data set in a MyOrders data table. The hierarchy data-coupling relation of the Quantity label control object and list line control object which were defined by the PropertyBinding object of a Quantity label control object establishes that the Text property of the label control object declared in the line 17 is data combined with the data item property identified as "Quantity". Hierarchy data-coupling relation specifies that a data item property is the property of the joint container of a Quantity label control object, i.e., the property of a MyOrders data table, again. Although the property of the MyOrders data table which is combined and serves as a Quantity label control object is a data item, any public properties of a server side data array (for example, a data table or a data object another type) may be combined with the

property of a server side control object including a data set index, data value size, a data value type, etc. [0068] A line 18 includes declaration of the server side table list control object which has a data-coupling data source property equal to "Page.MyItems." Therefore, MyItems is declared as a table list control object and its lower layer data source. In-line template declaration of lines 19-22 declares one or more child control objects of the table list control object declared in the line 18. The template instance relevant to in-line template declaration is used for a table list control object (namely, another repeat control object), and it creates and initializes the new instance of 0 to M of a suitable template control class. The number of the created new instances is controllable by the corresponding number of data sets in a data source.

[0069] The parameter of a template control object is defined in the lines 20-21 which show the list line control object containing the Item label which displays the text of a data item property from an Index label (that is, it corresponds to the sequence index of a given data set), and a data source. A list line control object is created to each data set of a data table MyItems under the table list control object declared with the line 18. The remaining lines in a file 600 contain the termination tag which closes declaration of the nest in a file.

[0070] The ASP+ resource 600 shows the declaration for performing one direction data coupling (that is, transmitting data to a control object hierarchy from a server side data table), and server side screen PUTINGU in an operation gestalt with this invention. Moreover, in another operation gestalt, one direction data coupling (that is, transmit data to a server side data table from a control object hierarchy) or the 2 direction data coupling (that is, transmit data to both directions) of an opposite direction can be declared.

[0071] In order to perform one direction data coupling of the 1st type in namely, the direction of a control object, the code in a line 7 makes a Load() subroutine an invalid, for example, loads the snapshot dump data array of the order from a MyOrderSystem data table to a local MyOrders data array (namely, one direction data coupling of the 1st above-mentioned type). Then, the property of a server side control object can be updated from the value from a local MyOrders data array between data-coupling phases according to the data-coupling statement of lines 14, 16, 17, 18, 20, and 21.

[0072] in order to perform one direction data coupling of the 2nd type from a control object namely, a page developer can make the Save() subroutine of a page object an invalid (see the preservation processing 516 of drawing 5). Before it, local MyOrders and a local MyItems data array are updated in the data-coupling phase using the value from a server side control object property according to the data-coupling statement of lines 14, 16, 17, 18, 20, and 21. Then, the Save() subroutine made into the invalid loads the value of a local-data array to the data table of a MyOrderSystem object. An example of the code declaration block which makes the Save() subroutine of a page object an invalid is shown in the lines 6-9 in the ASP+ resource 602 of drawing 6 B. This type of one direction data coupling can be used for loading an order to a server side database. In order to declare the 2 direction data coupling, a page developer can make an invalid both Load() of a page object, and a Save() subroutine.

[0073] Drawing 7 shows some control object hierarchies of a result who processed an example (for example, ASP+ resource) of the dynamic contents resource of drawing 6 A in 1 operation gestalt of this invention. The page control object and the various literal control objects of the best level are omitted from the control object hierarchy 700. The form control object 702 corresponding to the form declaration started from the line 11 of drawing 6 A contains the table list control object 704. The table list control object 704 will be started and declared from the line 12 of drawing 6 A, and will have a data-coupling property equal to a MyOrders data table.

[0074] the table list control object 704 contains the list line control objects 706-708 from ListRow0 to ListRowN namely, . It depends for the actual number of the list line control objects in the control object hierarchy in this level on the number of the data sets in a MyOrders data table. The configuration of each list line control object is defined by the item template declaration started with the line 13 of drawing 6 A. The data-coupling relation of each list line control object is defined in relation to the data source of the joint container (namely, table list control object 704) of each list line control object. In a certain operation gestalt, a control object joint container is the control object of high level rather than it is in the

nearest place in the control object hierarchy who supports an IBindingContainer interface.

[0075] The label control objects 710 and 712 corresponding to declaration of the lines 14 and 15 of drawing 6 A are contained in the joint container list line object 706. The data-coupling relation of the label control objects 710 and 712 is defined as the data set of a MyOrders data table in relation to the joint container list line control object 706 which is carrying out data coupling.

[0076] The table list control object 714 corresponding to the declaration in the line 16 of drawing 6 A is also contained in the joint container list line control object 706. The table list control object 714 is a joint container declared as data combined with MyOrders.MyIngredients. of a subdata table. The table list control object 714 contains the list line control objects 716-718 corresponding to the data set of subdata table MyOrders.MyIngredients. (from ListRow0 to ListRowN). Each to the list line control objects 716-718 is defined by the template declaration started from the line 19 of drawing 6 A. The data-coupling relation of the list line control objects 716-718 is defined in relation to the joint container table list control object 714.

[0077] Each to the list line control objects 716-718 contains a label control object (for example, 720 and 722). For example, the label control object 720 is declared as what carried out data coupling to the index field of the data set of subdata table MyOrders.MyIngredients in the line 20 of drawing 6 A, and the label control object 722 is declared as what carried out data coupling to the Ingredients data item of subdata table MyOrders.MyIngredients in the line 21 of drawing 6 A. In order to judge the data source of an Ingredients data item, the property of the label control objects 720 and 722 returns the reference to the nearest joint container (namely, control object which is an IBindingContainer instance like the list line control object 716) in the hierarchy of higher level.

[0078] Since a 708 or less list line control object hierarchy is defined by the same template declaration, he follows the same configuration as a 706 or less list line control object hierarchy. This hierarchy contains the 1st level containing the label control object 730, the label control object 734, and the table list control object 732. The list line control objects 736-738 of 0 to N are contained in the hierarchy of other level below the table list control object 732. Each of the list line control objects 736-738 contains a label control object which is defined by corresponding template declaration (see the label control objects 742, 744, 746, and 748).

[0079] Drawing 8 expresses the notation of an example of the server side control class in 1 operation gestalt of this invention. The server side control class defines the method common to all the server side control objects in an operation gestalt with this invention, the property, and the event. The more concrete control class (for example, server side carbon button control object corresponding to the client side carbon button in a web page) is derived from this control class. The control class 800 is shown as a thing containing the memory which stores a property 802 and a method 804. The operation gestalt of other control classes from which the combination of a data member and a method differs is also considered within the limits of this invention.

[0080] In the shown operation gestalt, the property 802 and the method 804 are public. Property "ID" is a string value in which reading which shows the control object identifier and writing are possible. Property "Visible" is a boolean value in which reading which shows whether the rendering of the authoring language data of a corresponding client side user interface element should be carried out and writing are possible. Property "MaintainState" is a boolean value in which reading which shows whether a control object should save the view state (and view state of the child object) at the time of termination of the present page request (by namely, response to the preservation processing 516 of drawing 5) and writing are possible. Property "Parent" is a reference in which reading by the control container object related in a control object hierarchy's current control object is possible. Property "Page" is a reference in which reading by the root page object to which it acts as the host of the current control object is possible. Property "Form" is a reference in which reading by the form control object to which it acts as the host of the current control object is possible. Property "Trace" is a reference in which reading by the trace context which enables the writing of a developer's trace log is possible. Property "BindingContainer" is a reference in which reading by the immediate-data joint container of a control object is possible. Property "Bindings" is a reference to the set of data-coupling association of a control

object.

[0081] A method 804 includes the access approach to processing of a request, and the data member of a control object. A method is referred in a certain operation gestalt by the pointer stored in the memory tooth space of a control object. This reference technique is used again also in the operation gestalt of a container control object, a control collection object, and other server side objects containing a page object. Method "Init()" is used for initializing these after a child control object is created (see the processing 504 of drawing 5). Method "Load()" is used for recovering view state information from a previous HTTP request (see the processing 506 of drawing 5). Method "Save()" is used for saving the view state information that it uses at a next HTTP request (see the processing 516 of drawing 5). Method "PreRender()" is used for performing a required pre rendering step in advance of preservation of a view state, and the rendering of contents (see the processing 514 of drawing 5). Method "Render (TextWriter output)" is used for outputting the authoring language code of the user interface element corresponding to the present control object (see the processing 518 of drawing 5). A code is outputted to TextWriter Output Screen passed to this code with the "Output" parameter. Before method "Dispose ()" ends a control object, it is used for doing a final clean-up activity (see the processing 520 of drawing 5).

[0082] Method "GetUniqueID()" obtains the string identifier as which the meaning of a current control object was recognized hierarchical. Method "GetControlWithID(String id)" returns the reference to the direct child control object which has the given identifier ("id"). Method "GetControlWithUniqueID (String id)" returns the reference to the child control object which has the hierarchy identifier ("id") of a meaning.

[0083] Method "PushControlPropertyTwoBindingContainer(String prop Name)" is an example of a push module, and if a postback value changes within a server side control object, it will be used for updating the joint container which carries out data coupling of two directions from postback data.

Method "PushBindingContainer PropertyTwoControl(String prop Name)" is an example of another push module, and is used for updating the server side control object property which has a current joint container value. Method "HasBindings()" returns the boolean value which shows whether the control object has joint association. These methods are used for solving joint relation with the property in the server side data array in relation to the property of a server side control object, and the server side data storage section (see the data-coupling processing 512 of drawing 5).

[0084] Drawing 9 shows the flow chart of the processing which carries out data coupling of the property of a server side control object to the property of a server side data array in an operation gestalt with this invention. Processing shown in drawing 9 can be performed in one or more processings shown in drawing 5 . For example, in an operation gestalt with this invention, processings 900 and 902 are performed between creation processing or the initialization processing 504, and processing 904 is performed between the load processings 506, and, as a result, a data table is stored as a property of the table list control object in processing 906. When a table list control object or its child control object is accessed while performing the postback input process processings 508 and 510 or data-coupling processing 512, a hierarchy is extended so that processings 908, 910, 912, and 914 may be performed and a joint container and its child may be included according to the configuration of a data table. Processings 916, 918, 920, and 922 are performed between the data-coupling processings 512 of drawing 5 , and perform processing 924 between preservation processings of drawing 5 . Although the above-mentioned relation corresponding to processing is used for the operation gestalt with this invention, it can replace or rearrange drawing 5 and sequence of processing of nine with other operation gestalten.

[0085] About each processing of drawing 9 , when the page is supporting one direction data coupling from a server side database, or when the 2 direction data coupling is being supported between a server side database and a control object hierarchy, processing 900 loads a local data table (namely, data array) from some server side databases using the function of COM+ reflection. Processing 902 associates a data table as a data source of a table list control object. In a certain operation gestalt, this correlation is carried out by the data-coupling property specified in declaration of a table list control object (see the

line 14 of drawing 6 A). Processing 904 creates a table list control object in a control object hierarchy. Processing 906 stores a data table as a property of a table list control object.

[0086] In processing 908, a table list control object is repeatedly referred to to each data set of a data table. Based on related template declaration, a table list control object creates a joint container control object (for example, list line control object) to each detected data set. Each joint container control object is treated as a child of a table list control object. Processing 910 relates each joint container control object to the data set of a data table by setting up the data-coupling property of a joint container control object so that it may become equal to the data set of a data source (for example, "MyOrders.Row(1)").

[0087] In processing 912, each joint container control object creates the lower layer control object corresponding to the property of the data set corresponding to a joint container of a data source. Or although each joint container control object creates a child control object, it will create the lower layer control object corresponding to the property of the data set corresponding to a joint container of a data source eventually. Processing 914 establishes the joint relation between the properties of a lower layer control object in the property of the response data set of a data source. The PropertyBinding object stored in the property of a lower layer control object can define this joint relation.

[0088] The decision processing 916 detects whether it should be transmitted to the joint container property with which the data from a control object property correspond. it should be transmitted -- ** -- when it detects, processing 918 calls the PushControl PropertyToBindingContainer() method in a lower layer control object, and transmits data to a joint container. Then, a procedure progresses to the decision processing 920. or the data from a control object property should be transmitted to the joint container property with which the decision processing 916 corresponds again -- ** -- also when not judging, a procedure progresses to the decision processing 920 from the decision processing 916, and the decision processing 920 judges whether data should be transmitted to the control object property which corresponds from a joint container property. it should be transmitted -- ** -- when it detects, processing 932 calls the PushBindingContainerPropertyToControl() method in a lower layer control object, and receives data from a joint container. Then, it progresses to processing 924. Or also when [by which the decision processing 920 should transmit the data from a joint container property to a control object property] it is judged that it does not come out, a procedure progresses to the decision processing 924 from the decision processing 920. When this page supports one direction data coupling to a server side database, or the 2 direction data coupling between a server side database and a control object hierarchy, processing 924 saves a local data table to a part of server side data storage section by COM+ reflection. Subsequently, a procedure continues to the rendering processing 518 of drawing 5 .

[0089] Drawing 10 shows the notation of an example of the server side property joint collection class in an operation gestalt with this invention. A server side property joint collection class defines the method and property of a property joint collection object (see the joint property of the server side control class of drawing 8). The property joint collection class 1000 is shown as a thing containing the memory stored in a property 1000 and a method 1004. The operation gestalt of other property joint collection classes from which the combination of a data member and a method differs is also considered to be within the limits of this invention.

[0090] In the displayed operation gestalt, the property 1002 and the approach 1004 are public. Property "All" is a snap shot dump array in which reading of a property joint object and dictation are possible, is set in order with an index and expresses all property association in a collection. a property -- "this [int index]" is a property joint object which returns the reference to the property joint object in the collection which was able to be set in order by the index and which can be read. Property "Count" is an integral value which shows the number of the property joint objects in a collection and which can be read.

[0091] A method 1004 includes the approach for accessing the data of a property joint collection object. Method"Add(PropertyBinding value)" is used for adding a specific property joint object to a collection. Method"IndexOf(PropertyBinding value)" is used in order to obtain the sequence index of the property joint object in a collection. Method"GetEnumerator(boolAllowRemove)" is used for obtaining ENYUMERETA (enumerator) of all property association in a collection. Method"Remove (PropertyBinding value)" is used for removing a specific property joint object from a collection.

Method "Remove(int index)" is used for removing the property joint object specified by the index from a collection. Method "Clear()" is used for removing all property joint objects from a collection.

[0092] Drawing 11 shows the notation of an example of the server side property joint class in an operation gestalt with this invention. A server side property joint class defines the method and property of a property joint object (see "All" of the property joint collection class of drawing 11 , and the property of "this [int index]"), and, thereby, defines the data-coupling relation between a control object and a server side data source. The property joint class 1100 is shown as a thing containing the memory which stores a property 1102. The operation gestalt of other classes from which the combination of a data member and a method differs is also considered to be within the limits of this invention.

[0093] In the illustrated operation gestalt, the property 1102 is public. Property "BindingContainer" is a joint container interface (namely, IBindingContainer object) in which reading and writing are possible, identifies a joint container data source and is used for obtaining access to there. An IBindingContainer object is a marker interface and identifies the control to which an own public property is made to use for the lower layer association. Therefore, a "BindingContainer" property offers the joint container identifier to which the combined control object discovers the server side data source of an association place, and makes access possible. Property "Property" is a string value (namely, property identifier) in which reading which identifies the property with which it was combined in the joint container object and writing are possible. Property "Control" is a control object in which reading and writing are possible (that is, the control object of the target with which a data source property is combined is identified). Property "ControlProperty" is a string value (namely, property identifier) in which reading which identifies the target property of a target control object and writing are possible. Property "FormatString" is a string value in which reading and writing are possible, and identifies an optional format concentration string.

[0094] The operation gestalt of this invention in this description is performed as a logic step of one or more computer systems. Logic processing of this invention is performed as a machine module with which it interconnected within the computer system beyond (2) 1 ** as a processor execute step sequence by which executive operation is carried out by the computer system beyond (1) 1 **. Activation is the problem of selection and it depends for it on the performance requirements of the computer system which performs this invention. Therefore, the logic processing which constitutes the operation gestalt of this invention of a publication on these descriptions can be variously expressed as processing, a step, an object, or a module.

[0095] A description, an above-mentioned example, and above-mentioned data offer the structure of the operation gestalt of this invention, and perfect explanation of an activity. Since this invention can be carried out with many gestalten, without deviating from the pneuma and the range of this invention, this invention is in an attached claim.

[0096]

[Effect of the Invention]

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] Generally especially this invention relates to the server side control object which combines the property of a server side data table in the two directions about a web server framework.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] A typical web browser receives data from the web server which specifies the appearance of a web page and basic actuation which are displayed in a client system. As a typical procedure, a user specifies the uniform resource (resource) locator (henceforth "URL") which is the global address of the resource on World Wide Web, and a desired website is accessed. Generally, the vocabulary "a resource" is the data which can be accessed by the program, or a routine. An example of URL is "HYPERLINK "http://www.microsoft.com/ms.htm" http://www.microsoft.com/ms.htm." The 1st part of this example of URL shows the given protocol (for example, "http") used for a communication link. The 2nd part specifies the domain name (for example, "HYPERLINK "http://www.microsoft.com" www.microsoft.com") which shows the location of a resource. The 3rd part specifies the resource in a domain (for example, file called "ms.htm"). This is followed and they are a browser and HYPERLINK "http://www.microsoft.com" www.microsoft.com. The HTTP (hypertext transport protocol) request relevant to the example of URL for taking out the data relevant to the ms.htm file in a domain is generated. The web server which is acting as the host of the www.microsoft.com site returns reception, the demanded web page, or a resource to a client system by the HTTP response, and displays a HTTP request on a browser.

[0003] The "ms.htm" file of the above-mentioned example contains the static HTML (HyperText Markup Language) code. HTML is a plain text authoring language used for the document (for example, web page) creation on World Wide Web. Since it is such, an HTML file can be taken out from a web server, can be displayed on a browser as a web page, and can offer the rich graphical experience which it comes to expect while the user is looking at the information from the Internet. A developer specifies the text and list which are displayed on a browser and by which formatting was carried out, form, a table, a hypertext link, an in-line image, voice, and background graphics using HTML. However, an HTML file is a static file which is not supporting dynamic generation of web page contents in essence.

[0004] When displaying dynamic contents, such as change of a stock price, and traffic information, the server side application program for treating a usually more complicated client-server dialogue is developed. A server side application program processes a HTTP request, and generates the suitable HTML code for transmission of the HTTP response to a client. An example of a HTTP request may contain parameters, such as data from the data or the web based form in an enquiry string. Since it is such, a server side application program can process this parameter, and can generate the HTML code dynamically by the HTTP response to a client. The document containing the suitable HTML code may be generated dynamically, using the sequence of one or more text write-in processings to a memory device by which formatting was carried out as an example of a server side application program. Then, it is transmitted to a client system by the HTTP response, and the obtained document is displayed on a browser as a web page there.

[0005] Development of a server side application program is a complicated activity as which it is required it is not only well versed in the usual HTML coding used for a web page design, but that it is well versed in pro GURAMUBE six including one or more programming language (for example, C++, Perl, Visual Basic, or Jscript). However, a web page architect is graphic designer or an editor in many

cases, and may be inexperienced in a program. Furthermore, if complicated web page development is simplified, the rate of development of new web contents can be gathered by any developers. Generally, a great effort is required and development of a custom-made server side application program is also actually like [which a developer regards as not wanting to often try it]. Therefore, it is desirable to offer the development framework to which a developer can create and process a web page dynamically by the minimum program.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any
damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention]

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] One means which makes min the requirements for a program of dynamic web page generation is an active server page (ASP) framework offered by Microsoft Corp. An ASP resource processes typically the HTTP request which specifies an ASP resource as a desired resource, after that, generates the HTML code as a result of the HTTP response to a client, for example, contains Visual Basic or Jscript. Furthermore, in order to mitigate a given application program effort, or the ASP resource was developed beforehand, refer to a third party's client side library component (for example, client side ACTIVEX control) for it. However, in a current server side application framework, programming required to manage dynamically client side user interface elements (for example, a text box, a list box, a carbon button, a hyperlink, an image, voice, etc.) within server side application needs a still advanced programming technique and a considerable effort. An unsolved technical problem is about processing a user interface element encapsulating programming demanded appropriately so that a web page developer can focus on other aspects of a web page.

[0007] A certain client side user interface element contains the data related with the server side data storage section like a database. An example of such an element is a client side table which displays the list of a product and prices from a server side product database. Moreover, the text box into which a consumer can input the shipping address which should be stored in a server side customer database as another example for next utilization is also included. The application program was written by the application developer, in order that a client side user interface might be written in a page by the page developer and might perform data communication between a page and a server side database by him conventionally. The problem is the point that both page development and application program development also requires a great development effort, and needs a database structure and the deep information about a certain amount of complicated programming (minding scree PUTINGU of a client side, or application programming by the side of a server).

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] According to this invention, **** and other technical problems are solved by offering the hierarchy data which combine one or more server side control objects and the one or more server side data storage sections. Hierarchy data coupling is supported by creating a joint container automatically to each data object in a data array. Each joint container offers access to the public property containing the data object of the server side data storage section associated to the child. By doing in this way, the property of a server side control object may be combined with the property of the server side data storage section through the joint container. Data-coupling relation is not limited to this, although it is establishable from the property of a server side control object in one direction to the property of the server side data storage section, or its hard flow. Although data-coupling relation is also establishable on the 2 direction targets between the property of a server side control object, and the property of the server side data storage section, it is not limited to this again.

[0009] In the means concerning a certain this invention, the approach of combining data with the property of the data set (for example, table line) of a server side data table which has one or more data sets for the property of a lower layer server side control object is offered in the server computer linked to a client computer system. This lower layer server side control object corresponds to a client side user interface element. A repeat control object is repeatedly referred to for every data set of a server side data table, and creates the joint container server side control object corresponding to each data set. Each joint container server side control object is related with the data set of a server side data table. A lower layer server side control object is created to the property of each data set. A given lower layer server side control object is in the lower layer of the joint container server side control object corresponding to a given data set, and is related with the property of a given data set. Data-coupling relation is established between one of the property of a lower layer server side control object, and the data sets of a server side data table.

[0010] In other means concerning this invention, a product is offered as a computer program product. In an operation gestalt with a computer program product, according to a computer system, reading is possible and the computer program storage which codes the computer program for performing the computer process which carries out data coupling of the property of a lower layer server side control object to the property of the server side data array which has one or more data objects is offered. As operation gestalt of a computer program product another again, it can also provide in the computer data signal which is materialized by the computer system by the subcarrier and codes a computer program. A child server side control object corresponds to a client side user interface element. A server side data array is loaded from a server side database. A repeat server side control object is related with a server side data table, and is created in a server side control object hierarchy. A server side data array is stored as a property of a repeat server side control object. One of the data objects and a corresponding joint container server side control object are created as a child of a repeat server side control object. A joint container server side control object is related with one of the data objects of a server side data array. A lower layer server side control object is created to each property of each data object. Each lower layer server side control object is in the lower layer of a joint container server side control object. Data-

coupling relation is established between the property of a lower layer server side control object, and the property of a server side data array.

[0011] In another means, the server which performs server side data coupling using the hierarchy containing the server side data array which has one or more data objects of the server side control object corresponding to a client side user interface element is also offered. Each data object may also contain a property. The repeat server side control object in a server side control object hierarchy is related with the server side data array. One or more joint container server side control objects are repeatedly created by the repeat server side control object based on the data object of a large number in a server side data array. One or more lower layer server side control objects correspond to the property of each data object of a server side data array. Each lower layer server side control is created by the lower layer of the combined-control server side control object of the given data object in a server side data array. Data-coupling relational structure describes the data-coupling relation between the property of a lower layer control object, and the property of the data object of a server side data array.

[0012]

[Embodiment of the Invention] In an operation gestalt with this invention, on a web server, web page contents are generated dynamically and displayed by the client. A client can become any browsers which support for example, the criterion HTML or other authoring languages. The client in a web server communicates through a network by using a HTTP request, a HTTP response, etc. Thus, although a web server is the HTML code gestalt, it generates web page contents and transmits the contents to a client. Thereby, a web page can be displayed on a browser. The server side control object which can respond to each user interface element and logic target of a web page is created on a web server, and processes and carries out the rendering of the web page contents. A server side control object is declared with dynamic contents resources, such as an ASP+ resource processed by the page factory which creates the hierarchy of a server side control object. The control object in a hierarchy collaborates and processes the request received from the client, subsequently, it generates result web page contents, transmits to a client, and terminates the control object in a hierarchy.

[0013] In an operation gestalt with this invention, a page object can be illustrated as a control object hierarchy's top level. In the page object which is also a control object, each child control object spreads to the hierarchy of a large number level including one or more child control objects including one or more child control objects in itself typically. A page object and a lower layer control object perform a processing sequence, and process and generate the web contents corresponding to a client side user interface element.

[0014] One of the processings in this sequence includes the data-coupling processing which controls the communication link with one or more server side control objects and server side data storage section. In an operation gestalt with this invention, the server side data storage section is a database which stores the data about the user interface element of a web page (see the data storage section 131 of drawing 1). Although not limited especially, other non-database mold data storage sections containing configuration registration, a data file, and a data stream are contained within the limits of this invention. By declaring a server side control object and the data-coupling relation of relation in a dynamic contents resource, a page developer can constitute an one direction and/or the control object hierarchy who has the 2 direction data-coupling relation to the server side data array relevant to the server side data storage section.

[0015] Drawing 1 shows the web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention. A client 100 performs the browser 102 which displays the web page 104 on the indicating equipment of a client 100. A client 100 may also include the client computer system which has a display like a video monitor. The "INTERNET EXPLORER" browser currently sold by Microsoft Corp. is an example of the browser 102 in an operation gestalt with this invention. As an example of other browsers, although there are "NETSCAPE NAVIGATOR", "MOSAIC", etc., it does not restrict to this. The table list 140 which has four list line elements is built into the illustrated web page 104. The list line element 142 is a child user interface element which the table list element 140 contains. The list line element 142 is defined by template declaration (see the

drawing 6 A as an example of template declaration), and has two trains containing the label elements 144 (for example, OrderID is expressed) and 146 (for example, the amount of an order product is expressed), and one train containing the table list element 148. Elements 144, 146, and 148 are declared as a child user interface element contained with the list line element 142 (see the drawing 6 A as an example of a label and table list declaration). The addition list line elements 150, 152, and 154 are also shown in the web page 104.

[0016] According to declaration of drawing 6 A, data coupling of the server side control object corresponding to the user interface element of a web page 104 is carried out to a server side [two] data array (for example, the server side data tables MyOrders and MyItems relevant to server side database access through a MyOrderSystem object). Generally, a server side data array contains an index data set (for example, a data line or a data object) in a control object hierarchy in relation to the server side data storage section (for example, it is loaded from the storing section and stored in/or the storing section). Such a data set has the property with which the data value of a certain table data line can be expressed, for example. The column 148 expresses the sub table list element which has two, two trains, i.e., the train of the line index from the Items data table by which (1) association was carried out, and train [of three list line elements which contain a label (for example, subject name) from (2) Items data table] **.

[0017] a browser 102 -- the HTTP response 112 -- setting -- a web server 116 to the HTML code -- reception -- things are made and the web page described by the HTML code is displayed. Although HTML is explained with reference to a certain operation gestalt, especially, it is not restricted and it is thought that it is the markup language of SGML (Standard Generalized Markup Language) and the XML base, and other authoring languages containing WML (Wireless Markup Language) designed so that the content and user interfaces of narrow-band radio equipment, such as a pocket bell (trademark) and a cellular phone, might be specified are within the limits of this invention. Furthermore, although the criterion HTML 3.2 is mainly indicated in this description, any versions of HTML may be contained within the limits of this invention.

[0018] The communication link with a client 100 and a web server 116 can be performed using a series of processings of the HTTP request 114 and the HTTP response 112. Although HTTP is explained with reference to a certain operation gestalt, it is not restricted especially and it is thought that other transport protocols including S-HTTP are within the limits of this invention. In a web server 116, the HTTP pipeline module 118 receives the HTTP request 114, analyzes URL, and calls the suitable handler 120 which processes a request. The web server 116 is equipped with two or more handlers 120 treating the resource of a different type in the operation gestalt with this invention.

[0019] For example, when URL specifies a static contents resource 122 like an HTML file, a handler 120 accesses the static contents resource 122, and transmits the static contents resource 122 to a client 100 by the HTTP response 112 through the HTTP pipeline 118. Or in an operation gestalt with this invention, when URL specifies a dynamic contents resource 124 like an ASP+ resource, a handler 120 accesses the dynamic contents resource 124, processes the contents of the dynamic contents resource 124, and generates the HTML code the result for web page 104. In an operation gestalt with this invention, the result HTML code contains standard HTML3.2 code. Generally, a dynamic contents resource is the server side declaration data storage section (for example, ASP+ resource) which can use the authoring language which describes the web page which should be displayed on a client for generating dynamically. And the HTML code for web pages passes the HTTP pipeline 118, and is sent to a client 100 by the HTTP response 112.

[0020] During this processing, the handler 120 was beforehand developed again, in order to make a development effort simplification, or it can access the library of the 3rd person code. One of such the libraries is the server side class control library 126, and a handler 120 can illustrate from here the server side control object which generates HTML data as a result of processing a user interface element and displaying on a web page. in an operation gestalt with this invention, one or more server side control objects are visible on the web page described by the dynamic contents file 124 -- it hides-like and maps to one or more user interface elements.

[0021] On the other hand, the 2nd library is a client side control class library 128 like the library

containing the "ACTIVEEX" component from Microsoft Corp. "ACTIVEEX" control is a COM (component object model) object which follows a fixed criterion in the method of a dialogue with a client and other components. Client side "ACTIVEEX" control is the component of the COM base as for which downloads to a client automatically and executive operation may be carried out to it by the web browser of a client. A server side ACTIVEEX component (not shown) is a component of the COM base which may be performed on a server, in order to achieve the various server side functions in which the server side functionality of stock price retrieval application or a database component is offered. ACTIVEEX is indicated by the detail by "understanding of ACTIVEEX and OLE" (David Chappelle, the Microsoft press, 1996).

[0022] In contrast with "ACTIVEEX" control, the server side control object of the operation gestalt of this invention specified as the dynamic contents resource 124 corresponds to the user interface element built into the web page on a client logically. A server side control object can generate the effective HTML code which may contain the locator which carries out the reference of the given client side "ACTIVEEX" control to for example, a HTML tag again. When the browser has already had the code for client side "ACTIVEEX" control in the storing system, executive operation of the "ACTIVEEX" control is carried out within the web page on a client. If it becomes right [that], a browser will download the code for "ACTIVEEX" control from the resource specified by the locator, and will carry out executive operation of the "ACTIVEEX" control within the web page on a client. The server side control object in the operation gestalt of this invention can start an event again to the server side "ACTIVEEX" control used for performing stock price retrieval application on a server.

[0023] A handler 120 accesses one or more non-user interface server components 130 which carry out executive operation on a web server 116 or another accessible web server again. A non-user interface server component 130 like stock price retrieval application or a database component is referred to in the dynamic contents resource 124 processed by the handler 120, or is related with it. The server side event started by the control object declared with the dynamic contents resource 124 may be processed in server side code which calls the suitable method of the non-user interface component 130. Consequently, the processing offered by the server side control object can simplify programming of non-user interface server component stereo-NENTO 130 by encapsulating processing and generation of a web page of a user interface element, and, thereby, the developer of the non-user interface server component 130 can be concentrated on development of the function of the application proper instead of a user interface problem.

[0024] In the operation gestalt with this invention, data coupling of the property of a server side control object is carried out to the property in the server side data storage section 131. For example, data coupling enables a page developer to relate the property of the control object in a server side control object hierarchy with the data item in a server side database. Each control object updates that data-coupling property during this processing by taking out the property value related from the data object in a database (it is called one direction data coupling from the server side data storage section). Or each control object updates the property of the server side data storage section again using a control object inner property value (it is called one direction data coupling to the server side data storage section). The 2 direction data coupling can be offered between a server side control object and the server side data storage section combining these two types of one direction data coupling.

[0025] Drawing 2 shows the flow chart of processing of the client side user interface element using the server side control object in an operation gestalt with this invention, and generation processing. In the transmitting processing 200, a client transmits a HTTP request to a server. A HTTP request contains URL which specifies resources, such as an ASP+ resource. In reception 202, a server calls the suitable handler which receives a HTTP request and processes the specified resource. Reading appearance of the ASP+ resource is carried out by the syntax-analysis processing 203. The creation processing 204 generates a server side control object hierarchy based on the content of the specified dynamic contents resource (for example, ASP+ resource).

[0026] In processing 206, a control object hierarchy's server side control object performs postback event handling, postback data handling, status management, and one or more processings in data coupling.

From a client, the postback event and data (collecting "postback input") from a user interface element are sent to a server, and are processed. Especially a postback event is not restricted and may also contain the "data change" event from the client side text box element sent to a "mouse click" event or a server from a client side carbon button element. Although postback data are not restricted to this, they may also contain the text inputted into the index of the item chosen from the text box element or the drop down box by the user.

[0027] In processing 208, each server side control object in a hierarchy is called in order to generate data like the HTML code for the display by the web page of a client side user interface element (or rendering). Although the vocabulary "a rendering" may mean the processing which displays graphics on a user interface, in this description, the "rendering" also means generation processing of the authoring language data which may be interpreted by client application like the browser for a display and a client side function. More detailed explanation of processing 206 and the rendering processing 208 is given in connection with drawing 6. In a certain operation gestalt, the call of the render() method in each control object is performed using a tree traversal sequence. That is, the call of the render() method of a page object becomes the repeat traverse covering the suitable server side control object in a hierarchy. As an option which calls the render() method of a suitable control object, approaches, such as event signaling or the object registration technique, may be used. A parenthesis specifies the "render()" label in which the method of comparing with a data value is shown.

[0028] In an operation gestalt with this invention, actual creation of each server side control object may delay until a server side control object is accessed in processings 206 or 208 (handling of a postback input, loading of a condition, rendering of a control object to the HTML code, etc.). Server processing is optimized when a server side control object is not accessed for a given request, and creation of a control object eliminates delay and unnecessary control object creation processing.

[0029] In the transmitting processing 210, the HTML code is transmitted to a client by the HTTP response. In reception 214, a client receives the HTML code relevant to the new web page which should be displayed. In display processing 216, a client system incorporates the user interface element of a new page according to the HTML code received from the HTTP response (for example, it displays).

however, a user interface element -- incorporating -- he should also understand that you may include non-display processing of control of offer of voice or a tactile-sense output, read-out to memory and writing, and script processing etc. A server side control object hierarchy is ended in a post process 212. In an operation gestalt with this invention, the server side control object in a hierarchy answers the HTTP request which refers to the associated ASP+ resource, is created, and after the rendering of authoring language data (for example, HTML data) finishes, it is destroyed. In another operation gestalt, processing 212 may be performed before processing 210 after processing 208.

[0030] Drawing 3 shows an example of the module in the web server used in an operation gestalt with this invention. A web server 300 accepts the HTTP request 302 in the HTTP pipeline 304. The HTTP pipeline 304 may also contain various modules, such as logging of web page statistics, user collating, a right of user access, and a module for the formation of an output cache of a web page. Eventually, each input HTTP request 302 received by the web server 300 is processed by the specific instance of an IHTTP handler class (it illustrates as a handler 306). A handler 306 analyzes a URL request and calls a suitable handler factory (for example, page factory module 308).

[0031] In drawing 3, the page factory module 308 related with the ASP+ resource 310 is called, and instantiation and configuration of the object of the ASP+ resource 310 are handled. In a certain operation gestalt, an ASP+ resource may be recognized by specifying a specific suffix (or a file extension part like ".aspx") as a file. If the request to a given ASP+ resource is first received by the page factory module 308, the page factory module 308 will search a file system, and will obtain a suitable file (for example, .aspx file 310). This file may also contain the text (for example, authoring language data) which may be behind interpreted or accessed by the server which processes a request, or the data (for example, cutting tool code data or coded data) in another format. When a physical file exists, the page factory module 308 reads a file to an aperture, and reads the file to memory. When a file is not found, the page factory module 308 returns the suitable error message "a file is not found."

[0032] After reading the ASP+ resource 310 to memory, the page factory module 308 processes a file content, and builds the data models (for example, a list [of script blocks], directive, static text area, and hierarchy server side control object, a server side control property, etc.) of a page. A data model is used for generating the source list of a new object class like the COM+ (Component Object Model+) class to which the class of the page base is made to extend. The page base class contains the code which defines the structure, property, and function of a basic page object. In an operation gestalt with this invention, it is compiled dynamically [a source list] subsequently to intermediate language, and is the instruction of a proper to a plat form behind. s (for example, X86, Alpha, etc.) It is compiled by timely (Just-In-Time). Intermediate language is COM+ IL. A code and Java A cutting tool code and Modula 3 A code and SmallTalk A general purpose or custom-made orientation linguistic code of a code, the Visual Basic code, etc. may also be included. In another operation gestalt, intermediate-language processing may be excluded and the instruction of a proper may be directly generated from a source list or a source file (for example, ASP+ resource 310). The control class library 312 may be accessed with the page factory module 308, in order to obtain the server side control class which is used for a control object hierarchy's generation and which was defined beforehand.

[0033] The page factory module 308 outputs the page object 314 which is a server side control object equivalent to the web page 104 of drawing 1 . The page object 314 and its child object are the control object hierarchy's 316 examples. It can think according to this invention, and like the custom-made control object, the example of other control objects is not restricted especially, either and also contains the object corresponding to HTML control of a table 1. The page object 314 corresponds to the web page 104 of drawing 1 logically, and relates to other control objects and hierarchy targets on a server. In a certain operation gestalt, a page object is a container object which contains the child control object hierarchical. The hierarchical relationship of other gestalten, such as a dependency, can be used with another operation gestalt. In a complicated control object hierarchy, one child object may be a container object of other child objects rather than it has the child object of a large number level (for example, the table list 140 is the container of the list line elements 142, 150, 152, and 154 and the child element of those). These server side control objects collaborate, handle the input from the HTTP request 302, manage the condition of a server side control object, perform data coupling, and carry out the rendering of the authoring language data (for example, HTML) for a web page as a result of a client. The result HTML code is an output from the control object hierarchy 316, and is transmitted to a client by the HTTP response 324.

[0034] It sets in the above-mentioned operation gestalt, the control object hierarchy's 316 control object is created and performed on a server 300, and each server side control object corresponds to the user interface element and logic target on a client. A server side control object collaborates, handles the postback input from the HTTP request 302 again, manages the condition of a server side control object, performs data coupling with a server side database, and generates the authoring language data (for example, the HTML code) used for the display of a web page as a result of a client. Result authoring language data are generated from the server side control object hierarchy 316 (namely, rendering), and are transmitted to a client by the HTTP response 324. For example, the result HTML code can embody any effective HTML configurations, and in addition to this, control of an ACTIVEX type, a JAVA (trademark) applet, a script, and when it is processed by the browser, refer to any web resources which produce client side user interface elements (for example, a control carbon button, a text box, etc.) for it.

[0035] By declaration made with the ASP+ resource 310, a server side control object can access one or more non-user interface server components 330 for a dialogue with the non-user interface server component 330 and a client side user interface element. For example, a postback input can be answered and a server side control object can start a server side event to the non-user interface server component registered into those events. Thus, the non-user interface server component 330 minds a user interface element for a dialogue with a user, and it can perform it, without programming a code required displaying and processing these elements.

[0036] Data coupling of the property of the one or more server side data storage sections 331 can be carried out to the property of a server side control object. Detailed explanation about a server side

control object hierarchy's example constituted so that one gestalt of data coupling might be supported is given in relation to drawing 7 . This data-coupling relation is declared in an ASP+ resource, it compiles a server side control object, and while carrying out run time activation, it is established.

[0037] In short, the operation gestalt of this invention contains on a server creation and the server side control object by which executive operation is carried out, in order to generate the HTML code transmitted to a client. The HTML code can embody any effective HTML configurations, and is control of an ACTIVEX type and JAVA, for example. Any web resources of the others which generate the user interface carbon button and other user interface elements in an applet, a script, and a client can be referred to. The user in a client can have a dialog with these user interface elements that correspond to a server side control object logically, and can return a request to a server. A server side control object is re-created on a server, and it processes the data, the event, and other properties of a user interface element so that the HTML code of the next round which should transmit to a client as a response may be generated.

[0038] If drawing 4 is referred to, an example of the computer system of the operation gestalt of this invention contains the general purpose computer equipment of the gestalt of the conventional computer system 400 containing the system bus 406 which connects the various system components containing the processor unit 402, a system memory 404, and a system memory 404 to the processor unit 400. System buses 406 may be any of the bus structure of some types containing the peripheral bus and local bus which use a memory bus or a memory controller, and various bus architectures. The system memory contains the memory (ROM) 408 only for playbacks, and random access memory (RAM) 410. The unformatted input / output system 412 (BIOS) containing the basic routine which helps a transfer of the information between the elements within a computer system 400 are stored in ROM408.

[0039] The computer system 400 contains the optical disk drive 418 which performs read-out and the writing of a removable optical disk 419 like the magnetic disk drive 414 and CD ROM which perform further read-out and the writing of a hard disk drive 412 and the removable magnetic disk 416 which perform read-out and the writing of a hard disk, DVD, or other optical media. The hard disk drive 412, the magnetic disk drive 414, and the optical disk drive 418 are connected to the system bus 406 by the hard disk drive interface 420, the magnetic disk drive interface 422, and the optical disk drive interface 424, respectively. A drive and its medium which can be related computer read offer the instruction of a computer system 400 which can be computer read, DS, a program, and the non-volatile storage section of other data.

[0040] Although the hard disk, the removable magnetic disk 416, and the removable optical disk 419 are used for this description in the above-mentioned environmental example of a publication, the medium type [other] in which data storage is possible and which can be computer read can be used for the above-mentioned example of a system. The medium of other types of these which can be used for the above-mentioned example of operating environment which can be computer read has for example, a magnetic cassette, flash memory card, a digital videodisc, a BERUNUI (Bernoulli) cartridge, random access memory (RAM), the memory (ROM) only for playbacks, etc.

[0041] Many program modules may be stored in a hard disk, a magnetic disk 416, an optical disk 419, and ROM408 or RAM410, and these contain the application program 428, other program modules 430, and the program data 432 of 426 or 1 or more operating systems. A user can input a command and information into a computer system 400 with input units, such as a keyboard 434 and a mouse 436, or other pointing equipments. As other input devices, there are a microphone, a joystick, a gamepad, a satellite dish, a scanner, etc., for example. These and other input devices are connected to the processor 402 through the serial port interface 440 connected to the system bus 406 in many cases. However, these input devices may be connected by other interfaces, such as a parallel port, a game port, or Universal Serial Bus (USB), again. The monitor 442 or the indicating equipment of other types is also connected with the system bus 406 through the interface of a video adapter 444 etc. In addition to a monitor 442, typically, a computer system contains other circumference output units (not shown), such as a loudspeaker and a printer.

[0042] A computer system 400 can operate in the environment using the logical connection to one or

more remote computers like a remote computer 446 connected by network. a remote computer 446 -- a computer system, a server, a router, Network PC, and a pier (peer) -- it may be equipment or other common network nodes, and there are many elements typically mentioned above in connection with a computer system 400, or all are included. Network connection contains Local Area Network (LAN) 448 and Wide Area Network (WAN) 450. Such a network environment is not new in office, an enterprise magnitude computer network, intranet, and the Internet.

[0043] When using by the LAN network environment, a computer system 400 is connected to a local network 448 through a network interface or an adapter 452. When using by the WAN network environment, a computer system 400 includes the modem 454 or other means for establishing the communication link by Wide Area Network 450 like the Internet typically. Built-in or external any is sufficient as a modem 454, and it is connected with the system bus 406 through the serial port interface 440. In the environment connected by network, the program module described in relation to the computer system 400 or its part may be memorized by remote memory storage. The illustrated network connection is an example and can use other means for the communication link establishment between computers.

[0044] In the operation gestalt of this invention, a computer 400 expresses a web server and a processor 402 carries out executive operation of the page factory module on the ASP+ resource memorized by at least one of storages 416, 412, 414, 418, and 419 or memory 404. A HTTP response and a request communicate by LAN448 connected to the client computer 446.

[0045] Drawing 5 is a process flowchart showing server side processing of the page object in an operation gestalt with this invention, and other control objects. In processing 500, the page object construction section is called with the page factory module 308 (refer to drawing 3). Consequently, a page object (see the page object 314 in drawing 3) is created so that it may correspond to the web page user interface element on a client logically. In processing 502, a page factory module calls the ProcessRequest member function of the page object which begins to process [gradual] the HTTP request received from the client. In the 1st phase of 1 operation gestalt of this invention, server side creation processing (not shown) creates the lower layer server side control object contained in the control object hierarchy of a page object. That is, in order that the construction section of a child control object may create a control object among the duration of HTTP request processing, call appearance is carried out repeatedly.

[0046] However, in another operation gestalt, creation of a child control object is delayable until a control object is needed for a given processing step (for example, rendering of the HTML code of the user interface element with which handling of a postback event, handling of postback data, loading of a view state and preservation, and data coupling analyze or correspond). Since the operation gestalt of the latter "which delays control object creation" can reduce utilization of unnecessary CPU and memory, it is the optimal. For example, there is a case where it will be called creation of the web page from which the user input event received from the client completely differs. In this case, a control object hierarchy is terminated promptly, and only in order to process the event which will illustrate a new different control object hierarchy of a new page, it is not necessary to illustrate the whole control object hierarchy of a former page.

[0047] The server call of the ProcessRequest method of a page object can be answered, and executive operation of the processings 504-520 can be carried out by the page object and each lower layer control object according to the data of a given HTTP request etc. In an operation gestalt with this invention, processings 504-520 are performed to each object of each in the sequence of drawing 5 . However, depending on a HTTP request, about given processing of another object, it may not be carried out in order or, as for the given processing to one object, processing may not be performed at all. For example, the 1st object performs the initialization processing 504 and its load processing 506, the postback data processing 508 is started and a lower layer control object performs the initialization processing 504 and the load processing 506 of itself by overdue control object creation after that. Although the sequence of processing by the page object and the lower layer control object is not restricted to this, it depends for it on whether the current condition of the property of the data in a HTTP request, a control object

hierarchy's configuration, and a control object and control object creation were overdue.

[0048] The initialization processing 504 initializes a control object, after a control object is created by carrying out executive operation of the server side code of the arbitration about initialization in a dynamic contents resource. Thus, each server side control object may be customized by the specific server side function declared with the dynamic contents resource. In an operation gestalt with this invention, a dynamic contents code customizes or extends the base page control class declared by the page developer with the ASP+ resource on a server. Compile of an ASP+ resource contains the declared code in the suitable initial code (for example, Init() method of a page object and a lower layer control object). The initialization processing 504 carries out executive operation of this code, and customizes or extends a page base class and the base class of a lower layer control object.

[0049] In an operation gestalt with this invention, the status management of a server side control object is supported in the load processing 506 and the preservation processing 516 in which the condition structure which can be conveyed is used, in order to contain the non-statement model of a client/server system by returning a server side control object to a former condition. With a certain operation gestalt, although a condition goes and comes back to a server with another next condition structure [in the one or more HTML fields in which the HTTP request / response of a couple hid] which can be conveyed, it is considered that other condition structures which can be conveyed are within the limits of this invention.

[0050] In a series of processings by the given present request and present given response about a page between a client and a server, the condition of one or more control objects is recorded on the condition structure which can be conveyed by the preservation processing 516 after processing of a previous request. In an operation gestalt with this invention, the status information of the addition containing the control object identifier which enables a hierarchy or a server to associate a suitable control object and a given condition is also included in the condition structure which can be conveyed. In the next HTTP request, status information is returned to a server with the condition structure which can be conveyed. A server extracts status information from the received condition structure which can be conveyed, loads condition data to the suitable control object in a control object hierarchy, and returns each control object to the condition that it existed before the previous HTTP response. After the processing to the present request, again, the condition of one or more server side control objects is recorded on the condition structure which can be conveyed by the preservation processing 516, and returns the condition structure which can be conveyed to a client by the next HTTP response by it.

[0051] Each server side control object is changed into the same condition as the condition before a previous HTTP request as a result of the load processing 506. For example, when a text box control object includes a property value equivalent to "JDoe" before a previous HTTP response, the load processing 506 returns the same control object to a previous condition by loading text string "JDoe" to the property value etc. Furthermore, whether the condition of a given object was memorized and recovered can also constitute.

[0052] When an operation gestalt with this invention is summarized, "it is saved" after the condition of one or more server side control objects processing. The saved status information is transmitted to a client by the response. A client returns the status information saved by the next response to a server side. A server loads status information to the newly illustrated server side control object hierarchy so that a hierarchy's condition may return to a former condition.

[0053] With another operation gestalt, status information can be held for some other accessible web locations by the server during round trip on a server of returning from a server to a client and a server. After a server receives a client request, this status information is taken out by the server and may be loaded to the suitable server side control object in a control object hierarchy.

[0054] In processing 508, the postback data received from the HTTP request are processed. Postback data may be contained in the pay load of the HTTP request by the key value and hierarchical display (for example, XML) which became a pair, or other data display like RDF (Resource Description Framework). Processing 508 analyzes the syntax of a pay load, and identifies the identifier of the meaning of a Target server side control object. When an identifier (for example,

"page:tablelist1:listrow2:label1") is found and a Target server side control object exists in a control object hierarchy, corresponding postback data are passed to the control object. For example, reference of drawing 1 sends the identifier and associated data of a meaning relevant to the data item of the list line 142 and a train 144 to a web server 116 with the pay load of the HTTP request 114. Processing 508 analyzes the syntax of the pay load of the HTTP request 114, and acquires the identifier and its related value (namely, the HTML code of a child user interface element like the list line 142) of a meaning of the table list 140. And processing 508 passes a postback value to the target control object which analyzes the identifier of the meaning of the list line 142, identifies and traverses and processes a corresponding server side control object.

[0055] As the load processing 506 was explained, the property value of a server side control object may be returned to a former condition. If postback data are received, a server side control object will judge whether the property value of the point to which the passed postback value corresponds was changed. When it changes, logging of the change is carried out to the change list in which data change of a related control object is shown. After all postback data are processed within a control object hierarchy, a control object method is called and one or more postdata change events can be started to one or more non-user interface server components like stock price retrieval application started on a server. A postback data change event is an event which shows that for example, postback data changed the property of a server side control object. In an instantiation operation gestalt, such an event is sent to a system offer event queue, and can call the server side code registered so that an event might be processed. Subsequently, a server side code can call the method of a non-user interface server component. Thus, a server side non-user interface server component can answer the event in which the trigger was carried out by change of the data of a server side control object. The option including using an application offer event queue, polling, and processing interruption which performs an event can also be considered to be within the limits of this invention.

[0056] A postback event is handled in processing 510. A postback event communicates with the pay load of a HTTP request. Processing 510 analyzes the syntax of the specific event target (for example, with the operation gestalt with this invention, label attachment is carried out with "#EVENTTARGET") which identifies the server side control object to which the event is turned. Furthermore, processing 510 analyzes the syntax of it, when there is a discovered event argument, and it gives it to the server side control object which had the event argument (for example, label attachment is carried out with "#EVENTARGUMENT" with the operation gestalt with this invention) specified. A control object starts the event processed with the non-user interface server component relevant to a dynamic contents resource.

[0057] Processing 512 solves the data-coupling relation between a server side control object and one or more server side databases with an accessible server, and updates a control object property with a database value by this, and updates the database field with the value of/or a control object property. the property of a server side control object may be related with the property of a parent data-coupling container as shown in the table of a server side application database with an operation gestalt with this invention (or data coupling is carried out -- having). A page frame work piece can update the control object property which has the value of the parent data-coupling container property corresponding to between the data-coupling processings 512 and by which data coupling was carried out. Thus, the user interface element on the web page in the next response reflects the updated property value in accuracy. The control object property with which a user interface element corresponds is because it was automatically updated between the data-coupling processings 512. Similarly, a control object property may be updated by the parent data-coupling container field, and, thereby, updates a server side application database in the postback input from a server side control object again.

[0058] The data-coupling relation of a given server control object and a given given data source (for example, server side data table) is declared in an ASP+ resource (see the line 14 of drawing 6 A). In order to support the common hierarchical property of joint data in a web page, data-coupling relation can support the hierarchy of itself (see the explanation of drawing 7). In a control object hierarchy, data-coupling relation identifies the property of the data table with which the property of a data table and a

server side control object is combined. Although data-coupling relation may be two directions again, this means that the property value of (1) server side data table may be loaded to the joint property of a server side control object, and the property value of (2) server side control object may be loaded to the joint property of a server side data table. All server side control objects reflect in accuracy the data-coupling property value appropriately updated from the server side data table after the data-coupling processing 512.

[0059] Processing 514 updates a large number by which executive operation may be carried out, before saving a control object condition and carrying out the rendering of the output. Processing 516 requires status information (namely, view state) from one or more control objects in a control object hierarchy, stores status information, and inserts it in the condition structure which is sent to a client with a HTTP response pay load and which can be conveyed. For example, a "grid" control object saves the present index page of the list of values so that a "grid" control object can return to this condition after the next HTTP request (namely, processing 506). As mentioned above, view state information expresses the condition of the control object hierarchy before the next action by the client. It will be used for changing a control object hierarchy into the condition of the point before client post back input process or data coupling if view state information returns.

[0060] The rendering processing 518 generates the suitable authoring language output (for example, HTML data) transmitted to a client by the HTTP response. A rendering is performed by top-down hierarchy tree WOKU and the embedded rendering code of all server side control objects. Processing 520 does the last clean-up activity (it is connection of a database in closing a file ****) of arbitration, and a control object hierarchy is terminated. Subsequently, processing performs return and processing 522 to 502, and a page object is ended by calling the destructive section there.

[0061] Drawing 6 A shows an example of a dynamic contents resource (for example, ASP+ resource) which declares a data-coupling control object to a server in the operation gestalt of this invention. A line 1 contains as a literal <html> which is the initiation tag of an HTML file. Up to lines 2-8, the data array of local "MyOrders" and "MyItems" expresses the code declaration block of a code which carries out a POPYU rate. Specifically, a line 2 declares the code by the side of a server as a server side script with a "runat=server" attribute and a value. This script makes an invalid load processing 506 of drawing 5 of a page object. Lines 3-5 declare an accessible public object in the name tooth space of a server side control object. "MyOrderSystem" of the "OrderSystem" type which is a public object is declared, this is used with a line 7, and a line 3 takes out the data array declared as "MyOrders" of a public data array object in the line 4 from the server side data storage section. It is used also for "MyOrderSystem" of a public object taking out the data array declared as "MyItems" of the public data array object in a line 5 from the server side data storage section again in a line 8.

[0062] A line 11 contains the literal which is the initiation tag <body> of the text of an HTML file. A line 12 contains the literal HTML code which specifies header "H1" which displays on a web page. A line 13 includes declaration of a server side form control object. A line 14 declares the server side table list control object which has a data-coupling data source property equal to "Page.MyOrders." Therefore, a MyOrders data table is declared as a table list control object and its lower layer data source.

[0063] In-line template declaration of lines 15-24 declares one or more child control objects of the table list control object declared with the line 14. Template declaration is interpreted at the time of syntax analysis, repeats a suitable ITemplate interface instance and combines it dynamically to the property of the same name in a control object (for example, table list control object). A table list control object (namely, repeat control object) uses an ITemplate instance at the time of activation, and creates and initializes the instance of N from 0 with a suitable new template control class. The number of the created new instances is controllable by the number of data sets (for example, data line) with which a data source corresponds.

[0064] For example, lines 15-24 declare a template called "itemtemplate" to the list line control object of the table list control object declared in the line 14. The parameter of a template control object is declared in lines 16-22, and shows the list line control object containing an OrderID label control object, a Quantity label control object, and other table list control objects. The table list control object declared in

the line 14 repeats and refers to the data set of a data table MyOrders, and creates a lower layer list line control object to each data set.

[0065] The list line control object corresponding to an itemtemplate template is developed by the control object developer as an IBindingContainer instance, therefore it is considered that it is a "joint container." Between the load processings 506 of drawing 5, the POPYU rate of the MyOrders data table is carried out by the code of a line 7 from MyOrderSystem. For example, the data offered by the MyOrdersSystem object are extracted from the server side database containing an order and product data. Then, a table list control object creates a list line control object as a joint container to each data set of a data table MyOrders.

[0066] The OrderID control object declared in the line 16 is combined with the OrderID property of the data set in a MyOrders data table. The hierarchy data-coupling relation of the OrderID label control object and list line control object which were defined by the PropertyBinding object of an OrderID label control object establishes that the Text property of the label control object declared in the line 20 is data combined with the data item property identified as "OrderID". Hierarchy data-coupling relation specifies that a data item property is the joint container of a control object, i.e., the property of a MyOrders data table, again. Although the property of the MyOrders data table which is combined and serves as an OrderID control object is a data item, any public properties of a server side data array (for example, a data table or a data object another type) are combinable with the property of a server side control object including a line index, data value size, a data value type, etc.

[0067] The Quantity control object declared in the line 17 is combined with the Quantity property of the data set in a MyOrders data table. The hierarchy data-coupling relation of the Quantity label control object and list line control object which were defined by the PropertyBinding object of a Quantity label control object establishes that the Text property of the label control object declared in the line 17 is data combined with the data item property identified as "Quantity". Hierarchy data-coupling relation specifies that a data item property is the property of the joint container of a Quantity label control object, i.e., the property of a MyOrders data table, again. Although the property of the MyOrders data table which is combined and serves as a Quantity label control object is a data item, any public properties of a server side data array (for example, a data table or a data object another type) may be combined with the property of a server side control object including a data set index, data value size, a data value type, etc.

[0068] A line 18 includes declaration of the server side table list control object which has a data-coupling data source property equal to "Page.MyItems." Therefore, MyItems is declared as a table list control object and its lower layer data source. In-line template declaration of lines 19-22 declares one or more child control objects of the table list control object declared in the line 18. The template instance relevant to in-line template declaration is used for a table list control object (namely, another repeat control object), and it creates and initializes the new instance of 0 to M of a suitable template control class. The number of the created new instances is controllable by the corresponding number of data sets in a data source.

[0069] The parameter of a template control object is defined in the lines 20-21 which show the list line control object containing the Item label which displays the text of a data item property from an Index label (that is, it corresponds to the sequence index of a given data set), and a data source. A list line control object is created to each data set of a data table MyItems under the table list control object declared with the line 18. The remaining lines in a file 600 contain the termination tag which closes declaration of the nest in a file.

[0070] The ASP+ resource 600 shows the declaration for performing one direction data coupling (that is, transmitting data to a control object hierarchy from a server side data table), and server side screen PUTINGU in an operation gestalt with this invention. Moreover, in another operation gestalt, one direction data coupling (that is, transmit data to a server side data table from a control object hierarchy) or the 2 direction data coupling (that is, transmit data to both directions) of an opposite direction can be declared.

[0071] In order to perform one direction data coupling of the 1st type in namely, the direction of a control object, the code in a line 7 makes a Load() subroutine an invalid, for example, loads the snap

shot dump data array of the order from a MyOrderSystem data table to a local MyOrders data array (namely, one direction data coupling of the 1st above-mentioned type). Then, the property of a server side control object can be updated from the value from a local MyOrders data array between data-coupling phases according to the data-coupling statement of lines 14, 16, 17, 18, 20, and 21.

[0072] in order to perform one direction data coupling of the 2nd type from a control object namely,, a page developer can make the Save() subroutine of a page object an invalid (see the preservation processing 516 of drawing 5). Before it, local MyOrders and a local MyItems data array are updated in the data-coupling phase using the value from a server side control object property according to the data-coupling statement of lines 14, 16, 17, 18, 20, and 21. Then, the Save() subroutine made into the invalid loads the value of a local-data array to the data table of a MyOrderSystem object. An example of the code declaration block which makes the Save() subroutine of a page object an invalid is shown in the lines 6-9 in the ASP+ resource 602 of drawing 6 B. This type of one direction data coupling can be used for loading an order to a server side database. In order to declare the 2 direction data coupling, a page developer can make an invalid both Load() of a page object, and a Save() subroutine.

[0073] Drawing 7 shows some control object hierarchies of a result who processed an example (for example, ASP+ resource) of the dynamic contents resource of drawing 6 A in 1 operation gestalt of this invention. The page control object and the various literal control objects of the best level are omitted from the control object hierarchy 700. The form control object 702 corresponding to the form declaration started from the line 11 of drawing 6 A contains the table list control object 704. The table list control object 704 will be started and declared from the line 12 of drawing 6 A, and will have a data-coupling property equal to a MyOrders data table.

[0074] the table list control object 704 contains the list line control objects 706-708 from ListRow0 to ListRowN namely,. It depends for the actual number of the list line control objects in the control object hierarchy in this level on the number of the data sets in a MyOrders data table. The configuration of each list line control object is defined by the item template declaration started with the line 13 of drawing 6 A. The data-coupling relation of each list line control object is defined in relation to the data source of the joint container (namely, table list control object 704) of each list line control object. In a certain operation gestalt, a control object joint container is the control object of high level rather than it is in the nearest place in the control object hierarchy who supports an IBindingContainer interface.

[0075] The label control objects 710 and 712 corresponding to declaration of the lines 14 and 15 of drawing 6 A are contained in the joint container list line object 706. The data-coupling relation of the label control objects 710 and 712 is defined as the data set of a MyOrders data table in relation to the joint container list line control object 706 which is carrying out data coupling.

[0076] The table list control object 714 corresponding to the declaration in the line 16 of drawing 6 A is also contained in the joint container list line control object 706. The table list control object 714 is a joint container declared as data combined with MyOrders.MyIngredients. of a subdata table. The table list control object 714 contains the list line control objects 716-718 corresponding to the data set of subdata table MyOrders.MyIngredients. (from ListRow0 to ListRowN). Each to the list line control objects 716-718 is defined by the template declaration started from the line 19 of drawing 6 A. The data-coupling relation of the list line control objects 716-718 is defined in relation to the joint container table list control object 714.

[0077] Each to the list line control objects 716-718 contains a label control object (for example, 720 and 722). For example, the label control object 720 is declared as what carried out data coupling to the index field of the data set of subdata table MyOrders.MyIngredients in the line 20 of drawing 6 A, and the label control object 722 is declared as what carried out data coupling to the Ingredients data item of subdata table MyOrders.MyIngredients in the line 21 of drawing 6 A. In order to judge the data source of an Ingredients data item, the property of the label control objects 720 and 722 returns the reference to the nearest joint container (namely, control object which is an IBindingContainer instance like the list line control object 716) in the hierarchy of higher level.

[0078] Since a 708 or less list line control object hierarchy is defined by the same template declaration, he follows the same configuration as a 706 or less list line control object hierarchy. This hierarchy

contains the 1st level containing the label control object 730, the label control object 734, and the table list control object 732. The list line control objects 736-738 of 0 to N are contained in the hierarchy of other level below the table list control object 732. Each of the list line control objects 736-738 contains a label control object which is defined by corresponding template declaration (see the label control objects 742, 744, 746, and 748).

[0079] Drawing 8 expresses the notation of an example of the server side control class in 1 operation gestalt of this invention. The server side control class defines the method common to all the server side control objects in an operation gestalt with this invention, the property, and the event. The more concrete control class (for example, server side carbon button control object corresponding to the client side carbon button in a web page) is derived from this control class. The control class 800 is shown as a thing containing the memory which stores a property 802 and a method 804. The operation gestalt of other control classes from which the combination of a data member and a method differs is also considered within the limits of this invention.

[0080] In the shown operation gestalt, the property 802 and the method 804 are public. Property "ID" is a string value in which reading which shows the control object identifier and writing are possible. Property "Visible" is a boolean value in which reading which shows whether the rendering of the authoring language data of a corresponding client side user interface element should be carried out and writing are possible. Property "MaintainState" is a boolean value in which reading which shows whether a control object should save the view state (and view state of the child object) at the time of termination of the present page request (by namely, response to the preservation processing 516 of drawing 5) and writing are possible. Property "Parent" is a reference in which reading by the control container object related in a control object hierarchy's current control object is possible. Property "Page" is a reference in which reading by the root page object to which it acts as the host of the current control object is possible. Property "Form" is a reference in which reading by the form control object to which it acts as the host of the current control object is possible. Property "Trace" is a reference in which reading by the trace context which enables the writing of a developer's trace log is possible. Property "BindingContainer" is a reference in which reading by the immediate-data joint container of a control object is possible. Property "Bindings" is a reference to the set of data-coupling association of a control object.

[0081] A method 804 includes the access approach to processing of a request, and the data member of a control object. A method is referred in a certain operation gestalt by the pointer stored in the memory tooth space of a control object. This reference technique is used again also in the operation gestalt of a container control object, a control collection object, and other server side objects containing a page object. Method "Init()" is used for initializing these after a child control object is created (see the processing 504 of drawing 5). Method "Load()" is used for recovering view state information from a previous HTTP request (see the processing 506 of drawing 5). Method "Save()" is used for saving the view state information that it uses at a next HTTP request (see the processing 516 of drawing 5). Method "PreRender()" is used for performing a required pre rendering step in advance of preservation of a view state, and the rendering of contents (see the processing 514 of drawing 5). Method "Render (TextWriter output)" is used for outputting the authoring language code of the user interface element corresponding to the present control object (see the processing 518 of drawing 5). A code is outputted to TextWriter Output Screen passed to this code with the "Output" parameter. Before method "Dispose ()" ends a control object, it is used for doing a final clean-up activity (see the processing 520 of drawing 5).

[0082] Method "GetUniqueID()" obtains the string identifier as which the meaning of a current control object was recognized hierarchical. Method "GetControlWithID(String id)" returns the reference to the direct child control object which has the given identifier ("id"). Method "GetControlWithUniqueID (String id)" returns the reference to the child control object which has the hierarchy identifier ("id") of a meaning.

[0083] Method "PushControlPropertyTwoBindingContainer(String prop Name)" is an example of a push module, and if a postback value changes within a server side control object, it will be used for updating

the joint container which carries out data coupling of two directions from postback data.

Method"PushBindingContainer PropertyTwoControl(String prop Name)" is an example of another push module, and is used for updating the server side control object property which has a current joint container value. Method"HasBindings()" returns the boolean value which shows whether the control object has joint association. These methods are used for solving joint relation with the property in the server side data array in relation to the property of a server side control object, and the server side data storage section (see the data-coupling processing 512 of drawing 5).

[0084] Drawing 9 shows the flow chart of the processing which carries out data coupling of the property of a server side control object to the property of a server side data array in an operation gestalt with this invention. Processing shown in drawing 9 can be performed in one or more processings shown in drawing 5 . For example, in an operation gestalt with this invention, processings 900 and 902 are performed between creation processing or the initialization processing 504, and processing 904 is performed between the load processings 506, and, as a result, a data table is stored as a property of the table list control object in processing 906. When a table list control object or its child control object is accessed while performing the postback input process processings 508 and 510 or data-coupling processing 512, a hierarchy is extended so that processings 908, 910, 912, and 914 may be performed and a joint container and its child may be included according to the configuration of a data table. Processings 916, 918, 920, and 922 are performed between the data-coupling processings 512 of drawing 5 , and perform processing 924 between preservation processings of drawing 5 . Although the above-mentioned relation corresponding to processing is used for the operation gestalt with this invention, it can replace or rearrange drawing 5 and sequence of processing of nine with other operation gestalten.

[0085] About each processing of drawing 9 , when the page is supporting one direction data coupling from a server side database, or when the 2 direction data coupling is being supported between a server side database and a control object hierarchy, processing 900 loads a local data table (namely, data array) from some server side databases using the function of COM+ reflection. Processing 902 associates a data table as a data source of a table list control object. In a certain operation gestalt, this correlation is carried out by the data-coupling property specified in declaration of a table list control object (see the line 14 of drawing 6 A). Processing 904 creates a table list control object in a control object hierarchy. Processing 906 stores a data table as a property of a table list control object.

[0086] In processing 908, a table list control object is repeatedly referred to to each data set of a data table. Based on related template declaration, a table list control object creates a joint container control object (for example, list line control object) to each detected data set. Each joint container control object is treated as a child of a table list control object. Processing 910 relates each joint container control object to the data set of a data table by setting up the data-coupling property of a joint container control object so that it may become equal to the data set of a data source (for example, "MyOrders.Row(1)").

[0087] In processing 912, each joint container control object creates the lower layer control object corresponding to the property of the data set corresponding to a joint container of a data source. Or although each joint container control object creates a child control object, it will create the lower layer control object corresponding to the property of the data set corresponding to a joint container of a data source eventually. Processing 914 establishes the joint relation between the properties of a lower layer control object in the property of the response data set of a data source. The PropertyBinding object stored in the property of a lower layer control object can define this joint relation.

[0088] The decision processing 916 detects whether it should be transmitted to the joint container property with which the data from a control object property correspond. it should be transmitted -- ** -- when it detects, processing 918 calls the PushControl PropertyToBindingContainer() method in a lower layer control object, and transmits data to a joint container. Then, a procedure progresses to the decision processing 920. or the data from a control object property should be transmitted to the joint container property with which the decision processing 916 corresponds again -- ** -- also when not judging, a procedure progresses to the decision processing 920 from the decision processing 916, and the decision processing 920 judges whether data should be transmitted to the control object property which

corresponds from a joint container property. it should be transmitted -- ** -- when it detects, processing 932 calls the PushBindingContainerPropertyToControl() method in a lower layer control object, and receives data from a joint container. Then, it progresses to processing 924. Or also when [by which the decision processing 920 should transmit the data from a joint container property to a control object property] it is judged that it does not come out, a procedure progresses to the decision processing 924 from the decision processing 920. When this page supports one direction data coupling to a server side database, or the 2 direction data coupling between a server side database and a control object hierarchy, processing 924 saves a local data table to a part of server side data storage section by COM+ reflection. Subsequently, a procedure continues to the rendering processing 518 of drawing 5.

[0089] Drawing 10 shows the notation of an example of the server side property joint collection class in an operation gestalt with this invention. A server side property joint collection class defines the method and property of a property joint collection object (see the joint property of the server side control class of drawing 8). The property joint collection class 1000 is shown as a thing containing the memory stored in a property 1000 and a method 1004. The operation gestalt of other property joint collection classes from which the combination of a data member and a method differs is also considered to be within the limits of this invention.

[0090] In the displayed operation gestalt, the property 1002 and the approach 1004 are public. Property "All" is a snap shot dump array in which reading of a property joint object and dictation are possible, is set in order with an index and expresses all property association in a collection. a property -- "this [int index]" is a property joint object which returns the reference to the property joint object in the collection which was able to be set in order by the index and which can be read. Property "Count" is an integral value which shows the number of the property joint objects in a collection and which can be read.

[0091] A method 1004 includes the approach for accessing the data of a property joint collection object. Method "Add(PropertyBinding value)" is used for adding a specific property joint object to a collection. Method "IndexOf(PropertyBinding value)" is used in order to obtain the sequence index of the property joint object in a collection. Method "GetEnumerator(bool AllowRemove)" is used for obtaining ENYUMERETA (enumerator) of all property association in a collection. Method "Remove (PropertyBinding value)" is used for removing a specific property joint object from a collection. Method "Remove(int index)" is used for removing the property joint object specified by the index from a collection. Method "Clear()" is used for removing all property joint objects from a collection.

[0092] Drawing 11 shows the notation of an example of the server side property joint class in an operation gestalt with this invention. A server side property joint class defines the method and property of a property joint object (see "All" of the property joint collection class of drawing 11, and the property of "this [int index]"), and, thereby, defines the data-coupling relation between a control object and a server side data source. The property joint class 1100 is shown as a thing containing the memory which stores a property 1102. The operation gestalt of other classes from which the combination of a data member and a method differs is also considered to be within the limits of this invention.

[0093] In the illustrated operation gestalt, the property 1102 is public. Property "BindingContainer" is a joint container interface (namely, IBindingContainer object) in which reading and writing are possible, identifies a joint container data source and is used for obtaining access to there. An IBindingContainer object is a marker interface and identifies the control to which an own public property is made to use for the lower layer association. Therefore, a "BindingContainer" property offers the joint container identifier to which the combined control object discovers the server side data source of an association place, and makes access possible. Property "Property" is a string value (namely, property identifier) in which reading which identifies the property with which it was combined in the joint container object and writing are possible. Property "Control" is a control object in which reading and writing are possible (that is, the control object of the target with which a data source property is combined is identified). Property "ControlProperty" is a string value (namely, property identifier) in which reading which identifies the target property of a target control object and writing are possible. Property "FormatString" is a string value in which reading and writing are possible, and identifies an optional format concentration string.

[0094] The operation gestalt of this invention in this description is performed as a logic step of one or more computer systems. Logic processing of this invention is performed as a machine module with which it interconnected within the computer system beyond (2) 1 ** as a processor execute step sequence by which executive operation is carried out by the computer system beyond (1) 1 **. Activation is the problem of selection and it depends for it on the performance requirements of the computer system which performs this invention. Therefore, the logic processing which constitutes the operation gestalt of this invention of a publication on these descriptions can be variously expressed as processing, a step, an object, or a module.

[0095] A description, an above-mentioned example, and above-mentioned data offer the structure of the operation gestalt of this invention, and perfect explanation of an activity. Since this invention can be carried out with many gestalten, without deviating from the pneuma and the range of this invention, this invention is in an attached claim.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention is shown.

[Drawing 2] The flow chart of processing for processing of a client side user interface element and a rendering is shown using the server side control object in an operation gestalt with this invention.

[Drawing 3] An example of the module of a web server used with an operation gestalt with this invention is shown.

[Drawing 4] An example of a useful system is shown in performing an operation gestalt with this invention.

[Drawing 5] The process flowchart showing processing of the page object in an operation gestalt with this invention is shown.

[Drawing 6 A] An example of a dynamic contents resource (for example, ASP+ resource) which declares the data-coupling control object in an operation gestalt with this invention is shown.

[Drawing 6 B] An example of a code declaration block which declares one direction data coupling of the 2nd type to the server side data table in an operation gestalt with this invention is shown.

[Drawing 7] Some control object hierarchies produced as a result of processing of the example of a dynamic contents resource (for example, ASP+ resource) of drawing 6 A in an operation gestalt with this invention are shown.

[Drawing 8] The notation of an example of the server side control class in an operation gestalt with this invention is shown.

[Drawing 9] The flow chart of data processing which combines the property of the server side control object in an operation gestalt with this invention and the property of a server side data array is shown.

[Drawing 10] The notation of an example of the server side property joint collection class in an operation gestalt with this invention is shown.

[Drawing 11] The notation of an example of the server side property joint class in an operation gestalt with this invention is shown.

[Description of Notations]

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

 WRITTEN AMENDMENT

----- [procedure amendment]
 [Filing Date] June 12, Heisei 13 (2001. 6.12)

[Procedure amendment 1]

[Document to be Amended] Description

[Item(s) to be Amended] 0096

[Method of Amendment] Modification

[Proposed Amendment]

[0096]

[Effect of the Invention] As mentioned above, according to this invention, offer the hierarchy data which combine one or more server side control objects and the one or more server side data storage sections, and the property of a server side control object minds the joint container. By making it combined with the property of the server side data storage section Processing a user interface element can encapsulate programming demanded appropriately. Not only about page development but about application program development It becomes possible to aim at relief of a development effort, without needing the deep information about a database structure or programming.

[Procedure amendment 2]

[Document to be Amended] Description

[Item(s) to be Amended] Explanation of a sign

[Method of Amendment] Modification

[Proposed Amendment]

[Description of Notations]

100 Client

102 Browser

104 Web Page

112 HTTP Response

114 HTTP Request

116 Web Server

118 HTTP Pipeline

120 Handler

122 Static Contents Resource

124 Dynamic Contents Resource

126 Server Side Control Class Library

128 Client Side Control Class Library

130 Non-UI Server Component

131 Server Side Data Storage Section

300 Web Server

304 HTTP Pipeline

308 Page Factory
310 ASP+ Resource
312 Control Class Library
314 Page Object
318 Text Box Object
320 322 Carbon button object
330 Application Component
331 Server Side Data Storage Section
400 Computer
402 CPU
404 Memory
412 Hard Disk Drive
414 Magnetic Disk Drive
416 Removable Storage
418 Optical Disk Drive
419 Optical Disk
420, 422, 424 I/F
426 Operating System
428 Application Program
430 Program Module
432 Program Data
434 Keyboard
436 Mouse
440 Serial Port Interface
442 Monitor
444 Video Adapter
446 Remote Computer
448 LAN
450 WAN
452 Network Adaptor
454 Modem
702 Form
704, 714, 732 Table list
800 Control Class
802, 1002, 1102 Property
804 1004 Method
1000 Property Joint Collection Class
1100 Property Joint Class

[Translation done.]